# An extension of the Linear Regression Model for Improved Vessel Trajectory Prediction utilising *a priori* AIS Information

by

**Christiaan Neil Burger**


Thesis presented in the partial fulfilment of the requirement for the degree of

*Master of Science (Computer Science)*

in the Faculty of Science at the University of Stellenbosch

Supervisor: Dr. Trienko Lups Grobler

Co-supervisor: Prof. Waldo Kleynhans

**April** 2022

# DECLARATION

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

April 2022

# ABSTRACT

## An Extension of the Linear Regression Model for Improved Vessel Trajectory Prediction - utilising *a priori* AIS Information

C.N. Burger

*Department of Mathematical Sciences, Division of Computer Science,*
*University of Stellenbosch,*
*Private Bag X1, Matieland 7602, South Africa*

Thesis: M.Sc (Computer Science)
April 2022

As maritime activities increase globally, there is a greater dependency on technology in monitoring, control and surveillance of vessel activity. One of the most prominent systems for monitoring vessel activity is the Automatic Identification System (AIS). An increase in both vessels fitted with AIS transponders, and satellite- and terrestrial receivers has resulted in a significant increase in AIS messages received globally. This resultant rich spatial and temporal data source related to vessel activity provides analysts with the ability to perform enhanced vessel movement analytics, of which a pertinent example is the improvement of vessel location predictions. In this thesis, we propose a novel method for predicting future locations of vessels by making use of historic AIS data. The proposed method extends a Linear Regression Model (LRM), utilising historic AIS movement data in the form of *a priori* generated spatial maps of the course over ground (LRMAC). The LRMAC has low complexity and is programmatically easy to implement, and attains accurate prediction results. We first compare the LRM with a Discrete Kalman Filter (DKF) on linear trajectories. We then extend the LRM to form the LRMAC. The LRMAC is compared to another method in literature called the Single Point Neighbour Search (SPNS). For the use case of predicting Cargo and Tanker vessel trajectories, with a prediction horizon of up to six hours, the LRMAC has an improved execution time and performance compared to the SPNS.

**Key words:**

Linear Regression Model, Automatic Identification System (AIS), Vessel Trajectory Prediction, Spatial Maps, Data Mining

# OPSOMMING

## 'n Uitbreiding van die Lineêre Regressiemodel vir Verbeterde Vaartuig-trajek Voorspelling - deur gebruik te maak van *a priori* OIS Inligting

C.N. Burger

*Departement van Wiskundige Wetenskappe, Divisie van Rekenaarwetenskap,*
*Universiteit van Stellenbosch,*
*Privaatsak X1, Matieland 7602 , Suid-Afrika.*

Tesis: M.Sc (Rekenaarwetenskap)
April 2022

As gevolg van die toename in maritieme aktiwiteite wêreldwyd, het die afhanklikheid van tegnologie in die monitering, beheer en toesig van vaartuigaktiwiteite ook toegeneem. Een van die mees prominente stelsels vir die monitering van vaartuigaktiwiteit is die Outomatiese Identifikasiestelsel (OIS). 'n Toename in vaartuie wat toegerus is met OIS-transponders, en die toename in satelliet- en terrestriële ontvangers, het gelei tot 'n aansienlike groei in OIS-boodskappe wat wêreldwyd ontvang is. Dit het weer gelei tot die toename in dataryke ruimte-temporele bronne, wat verband hou met vaartuigaktiwiteite. Dit gee ontleders die vermoë om gevorderde vaartuig-bewegingsanalise uit te voer, waarvan 'n toepaslike voorbeeld, die verbetering van vaartuig-liggingvoorspelling is. In hierdie tesis stel ons 'n nuwe strategie voor om toekomstige liggings van vaartuie te voorspel, wat gebruik maak van historiese OIS-data. Die voorgestelde metode brei 'n Lineêre Regressie Model (LRM) uit, deur gebruik te maak van historiese bewegingsdata en ruimte kaarte van *a priori* koers oor grond inligting (LRMAK). Die LRMAK het 'n lae kompleksiteit en is programmaties eenvoudig om te implementeer, met relatiewe akkurate voorspelling resultate. Ons vergelyk eers die LRM met 'n Diskrete Kalman Filter (DKF) op lineêre trajekte. Dan brei ons die LRM uit om die LRMAK te vorm. Die LRMAK word vergelyk met 'n ander metode in literatuur wat die Enkel-punt Buursoektog (EPBS) genoem word. In die geval van trajek-voorspelling vir vrag- en tenkwa-vaartuie, het die LRMAK 'n verbeterde uitvoeringstyd en is vergelykbaar met 'n ander algoritme in literatuur, die EPBS, tot en met 'n voorspellingstydperk van ses-ure.

**Sleutelwoorde:**
Lineêre regressiemodel, outomatiese identifikasiestelsel ( OIS), v aartuigtrajekvoorspelling, ruimte kaarte, data-ontginning

# ACKNOWLEDGEMENTS

First, I'd like to thank **my parents**. Thank you for always being there and supporting me from the day I was born in everything I did. I am who I am because of you. Thank you for making it possible to pursue my dreams and passions. I love you both very much!

Thanks to all my **grandparents**, even though you all have passed, you forever live in my heart. You helped form my love for learning and inspired me to become the man I am today.

I want to thank my supervisors, **Dr. Trienko L. Grobler** and **Prof. Waldo Kleynhans** for your guidance, insights, knowledge, inspiration, and time over the years. It will remain with me for the rest of my life. Thank you **Dr. Trienko L. Grobler** for being available at almost any time of day when I needed help, and thank you for your patience.

I want to thank all of my **teachers** over the years. You inspired me and helped form my love of learning.

Thanks to my **friends and family**, although your names are not explicitly mentioned, I appreciate you all very much! Thank you for being there and supporting me when the times were tough.

Thanks to my brother **Duard Burger**, for all the sketches/illustrations used in this thesis!

Last but not least, thanks to **MUNUS International** for the financial support during the duration of my M.Sc degree, and an extra special thanks to the Director, **Prof. Waldo Kleynhans**, for all your support!

Never quit. Keep on going. We, as humans, constantly face new challenges every day. As my friends and family know, I have a favourite quote. I want to thank the creator thereof and share the quote: "Tough times never last, only tough people last." - Baba Musia (aka Demi Demi)

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF APPENDICES

# ACRONYMS

**AE** autoencoder

**AI** Artificial Intelligence

**AIS** Automatic Identification System

**ANN** Artificial Neural Network

**ASP** Application Service Providers

**Bi-GRU** Bidirectional Gated Recurrent Unit

**BSA** Binary search algorithm

**CN** close neighbour

**CNN** Convolutional Neural Network

**COG** Course over Ground

**COG SD** COG standard deviation

**COGSM** Course Over Ground SM

**CPS** Communication Service Providers

**CR** Coastal Radar

**CRS** Coastal Radar System

**CSTDMA** Carrier Sense Time-Division Multiple Access

**DBSCAN** Discovering Clusters in Large Spatial Databases with Noise

**DC** Date Centres

**DDP** Data Distribution Plan

**DKF** Discrete Kalman Filter

**DL** Deep Learning

**DSC** Digital Selective Calling

**ECDIS** Electronic Chart Display and Information System

**EEZ** Exclusive Economic Zones

**EKF** Extended Kalman Filter

**EM** Expectation Maximisation

**EMSA** European Maritime Safety Agency

**ENC** Electronic Navigation Charts

**FCNN** Fully-Connected Convolutional Neural Network

**GAN** Generative Adversarial Network

**GB** Gigabyte

**GMM** Gaussian Mixture Models

**GP** Gaussian Process

**GPS** Global Positioning System

**GRU** Gate Recurrent Unit

**GT** Gross Tonnage

**HF** High Frequency

**HFR** High Frequency Radar

**IDE** International Data Exchange

**IMO** International Maritime Organisation

**KB-PF** Knowledge Based Particle Filter

**KB-VM** Knowledge Based Velocity Model

**kt** knots

**LAT** Latitude

**LON** Longitude

**LRIT** Long Range Identification and Tracking

**LRM** Linear Regression Model

**LRMAC** Linear Regression Model with *a priori* COG information

**LS** Least-Squares

**LSSVR** Least-Squares Support Vector Regression

**LSTM** Long Short-Term Memory

**MAPE** Mean Absolute Percentage Error

**MED** Mean Euclidean Distance

**MF** Medium Frequency

**MICE** Multivariate Imputation by Chained Equations

**MID** Maritime Identification Digits

**ML** Machine Learning

**MLSSVR** Multiple outputs Least-Squares Support Vector Regression

**MMSI** Maritime Mobile Service Identity

**MP-LSTM** Multistep Prediction Long Short-Term Memory

**MSA** Maritime Situational Awareness

**NCDM** Neigbor Course Distribution Method

**NM** Nautical Miles

**NN** Neural Network

**OU** Ornstein-Uhlenbeck

**pdf** probability density function

**PE** predictor equations

**PF** Particle Filter

**PreMovEst** Select Best AIS Data in Prediction Vessel Movements and Route Estimation

**PSSP** Point-based Similarity Search Prediction

**QGIS** Quantum Geographic Information System

**RDF** Radio Direction Finder

**RF** Random Forest

**RMSE** Root Mean Square Error

**RNN** Recurrent Neural Network

**RS** Radar System

**S-AIS** Satellite based Automatic Identification System

**SAR** Synthetic Aperture Radar

**SARTs** Search and Rescue Transponders

**SD** standard deviation

**SDM** spatial distribution map

**SM** Spatial Map

**SM-OMLSSVR** Online Multiple outputs Least-Squares Support Vector Regression model based on a Selection Mechanism

**SOG** Speed over Ground

**SOLAS** Safety of Life at Sea

**SOTDMA** Self Organised Time Division Multiple Access

**SPNS** Single Point Neighbour Search

**STENet** Ship Traffic Extraction Network

**SVR** Support Vector Regression

**T-AIS** Terrestrial based Automatic Identification System

**THREAD** Traffic Route Extraction and Anomaly Detection

**TPNet** Trajectory Proposal Network for Motion Prediction

**TSSP** Trajectory-based Similarity Search Prediction

**TSSPL** Trajectory-based Similarity Search Prediction based on a RNN LSTM

**UCT** Coordinated Universal Time

**UK** United Kingdom

**UN** United Nations

**UNCLOS** United Nations Convention on the Law of the Sea

**US** United States

**UTM** Universal Transverse Mercator

**VHF** Very High Frequency

**VMS** Vessel Monitoring System

**VTS** Vessel Tracking Service

**WGS-84** World Geodetic System - 1984

# NOMENCLATURE

### Matrices

| | |
|---|---|
| $\boldsymbol{\Psi}$ | COG Spatial Map |
| $\boldsymbol{\Sigma}$ | COG standard deviation spatial map |
| $\boldsymbol{A}$ | DKF state transition matrix |
| $\boldsymbol{B}$ | DKF output matrix |
| $\boldsymbol{H}$ | DKF transformation matrix |
| $\boldsymbol{K}$ | Vessel Counts Spatial Map |
| $\boldsymbol{K}_t$ | Kalman gain at time step $t$ |
| $\boldsymbol{P}_0$ | DKF initial error covariance matrix |
| $\boldsymbol{P}_{t-1}$ | DKF update error covariance estimate matrix |
| $\boldsymbol{P}_t$ | DKF updated error covariance estimate |
| $\boldsymbol{P}_t^-$ | DKF prediction error covariance estimate matrix |
| $\boldsymbol{Q}$ | DKF process noise |
| $\boldsymbol{R}$ | DKF observational noise covariance matrix |
| $\boldsymbol{X}$ | The matrix with all the historic AIS data observations, with respect to the SPNS |

### Number sets

| | |
|---|---|
| $\mathbb{N}$ | Natural Numbers |
| $\Re$ | Real Numbers |

### Other Symbols

| | |
|---|---|
| $\#T_h$ | Total number of subtrajectories generated from $\boldsymbol{T}_i$ of length $h$ |
| $\bar{\chi}_c$ | Predicted COG from the CN set |
| $\bar{c}$ | The mean cosine of the COG of the CN set |
| $\bar{l}$ | Average number of metres that one degree of latitude and longitude span |

| | |
|---|---|
| $\bar{s}$ | The mean sine of the COG of the CN set |
| $\mathbf{1}_{\boldsymbol{\Psi}}$ | Indicator function that sets $\rho$ to zero |
| $\mathbf{1}_{n_t>\omega}$ | Indicator function |
| $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ | Spatial Map index location at the predicted $\phi$ and $\lambda$ |
| $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ | Spatial Map index location vector at the predicted $\phi$ and $\lambda$ |
| $\chi_i$ | COG of observation $i$ |
| $\Delta\chi$ | Maximum allowable COG deviation of the CN set |
| $\Delta k_t$ | Prediction time step between two subsequent predictions |
| $\Delta l$ | SPNS prediction step length |
| $\delta_{s_i}$ | Element in $\boldsymbol{\delta}_s$ at index $i$ |
| $\eta$ | Neighbourhood parameter |
| $\hat{\boldsymbol{\psi}}_{t+1}$ | Predicted COG by the LRMAC |
| $\hat{\chi}_i^{k+}$ | Predicted COG from the CN set |
| $\hat{V}_{\omega,c_{t-1}}$ | Estimated $y$-intercept of the LRM |
| $\hat{V}_{\omega,t}$ | Predicted SOG using all observations up until time-step $t-1$ |
| $\hat{v}_i^{k+}$ | Predicted SOG from the CN set |
| $\hat{x}$ | Estimated $x$-coordinate |
| $\hat{y}$ | Estimated $y$-coordinate |
| $\iota$ | Interval starting point |
| $\kappa$ | Length of a cell of the Spatial Map |
| $\lambda$ | Longitude |
| $\lambda_{s_i}$ | Element in $\boldsymbol{\lambda}_s$ at index $i$ |
| $\mathcal{O}$ | Denotes Big-$\mathcal{O}$ notation |
| $\nabla\hat{V}_{\omega,t-1}$ | Estimated gradient of the LRM |

$\omega$ Window size

$\phi$ Latitude

$\psi$ True COG recorded by a vessel, in degrees (°)

$\rho$ Confidence measure (scaling factor)

$\sigma_x$ Initial $x$-coordinate variance for the DKF error covariance matrix $\boldsymbol{P}_0$

$\sigma_x^V$ Initial $x$-axis acceleration variance for the DKF error covariance matrix $\boldsymbol{P}_0$

$\sigma_y$ Initial $x$-coordinate variance for the DKF error covariance matrix $\boldsymbol{P}_0$

$\sigma_y^V$ Initial $y$-axis acceleration variance for the DKF error covariance matrix $\boldsymbol{P}_0$

$\text{MED}_{\boldsymbol{T}_i,\lambda_{s_i}}$ Mean Euclidean Distance of trajectory $\boldsymbol{T}_i$ and undersample rate $\lambda_{s_i}$

$\text{MED}_{\lambda_{s_i}}$ Average MED of undersample rate $\lambda_{s_i}$

$\tilde{v}_c$ Median Speed of the CN set

$\varphi_t$ ROT of a vessel at time-step $t$

$a_x$ DKF acceleration in the x direction $(m/s^2)$

$a_y$ DKF acceleration in the y direction $(m/s^2)$

$C_n$ The number observations in the CN set

$d(\hat{\boldsymbol{p}}^k, \boldsymbol{p}_i)$ Haversine distance between the LON and LAT coordinates of $\hat{\boldsymbol{p}}^k$ and $\boldsymbol{p}_i$

$d_l$ Number of dataset features

$f(\hat{\phi}^k)$ Function of $\phi$ to transform LON from metres to degrees

$g(\hat{\phi}^k)$ Function of $\phi$ to transform LAT from metres to degrees

$h$ Prediction length

$i_\lambda$ Index value at $\lambda$ with respect to the Spatial Map grid

$i_\phi$ Index value at $\phi$ with respect to the Spatial Map grid

$J^{max}$ Maximum number of nodes at any given level in a prediction tree

$K^s$ Number of predicted positions a trajectory consists of

| | |
|---|---|
| $k_i$ | Total time that has elapsed after having recorded the $i^{\text{th}}$ true observation |
| $k_t$ | The elapsed time in seconds at time-step $t$ |
| $L$ | Number of AIS states (observations) recorded in the dataset |
| $M$ | Indicates the number of AIS messages recorded |
| $n$ | Number of observations |
| $n_d$ | Maximum number between the number of rows or columns of the Spatial Map |
| $n_f$ | Length of the state vector for the DKF |
| $n_m$ | Number of observations used to estimate the least squares parameters |
| $n_s$ | Number of observations present in the dataset used by the SPNS |
| $n_t$ | Total number of true observations that were recorded after $\Delta k_t \cdot t$ seconds |
| $n_{\boldsymbol{\lambda}}$ | Number of observations in $\boldsymbol{\lambda}_s$ |
| $n_\delta$ | Number of observations in $\boldsymbol{\delta}_s$ |
| $n_{\hat{\lambda}}$ | Spatial Map index of the predicted LON |
| $n_{\hat{\phi}}$ | Spatial Map index of the predicted LAT |
| $n_{d_c}$ | Number of columns in of a Spatial Map |
| $n_{d_r}$ | Number of rows in of a Spatial Map |
| $n_{l_c}$ | Number of columns in an Spatial Map |
| $n_{l_r}$ | Number of rows in an Spatial Map |
| $n_l$ | Number of observations used to estimate the least squares parameters |
| $r_c$ | SPNS search radius for the CN set |
| $S$ | Interval of course angles considered by the SPNS to include observations in the CN set |
| $s$ | Stride starting position |
| $t$ | Time step |
| $t_i$ | The timestamp at observation $i$ |

$V_i$ True $i^{\text{th}}$ recorded SOG of a vessel

$V_t'$ SOG recorded by the AIS transmitter

$V_t$ SOG converted to $m/s$, and in terms of the LRMAC $^\circ/s$

$x_t$ Recorded $x$-coordinate of a vessel (UTM)

$y_t$ Recorded $y$-coordinate of a vessel (UTM)

**Values/subsets extracted from Matrices**

$\boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$ *a priori* average COG at index location $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$

$\boldsymbol{\Psi}_{i_\phi,i_\lambda}$ Mean COG value at the associated indices of $\phi$ and $\lambda$ in the respective Spatial Map grid

$\boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$ COG SD at the corresponding index location $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$

$\boldsymbol{\Sigma}_{i_\phi,i_\lambda}$ COG standard deviation value at the corresponding index locations of $\phi$ and $\lambda$ in the respective spatial map grid

$\boldsymbol{K}_H$ *A priori* cell counts matrix at the index locations in $\boldsymbol{H}$

$\psi_{(i_\phi,i_\lambda)_j}$ The $j^{\text{th}}$ COG value at the respective index locations of $\phi$ and $\lambda$ in the respective Spatial Map grid

$\psi_{i_\phi,i_{\lambda_j}}$ The $j^{\text{th}}$ recorded COG value at index values $i_\phi$ and $i_\lambda$ in the Spatial Maps

**Vectors / Lists / Arrays**

$\boldsymbol{\alpha}$ An array of undersample sets

$\boldsymbol{\delta}_s$ Set of prediction horizons

$\boldsymbol{\epsilon}_{\text{LRMAC}}$ Jagged array of LRMAC prediction errors

$\boldsymbol{\epsilon}_{\text{SPNS}}$ Jagged array of SPNS prediction errors

$\hat{\boldsymbol{T}}_i$ Predicted trajectory of a vessel

$\boldsymbol{\lambda}_s$ Set of undersample rates

$\boldsymbol{\Lambda}_t$ Cosine and Sine of the COG used to determine the SOG in the respective LAT and LON directions

| | |
|---|---|
| $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ | Spatial Map index location |
| $\boldsymbol{p}_i^T$ | The position vector denoting $\lambda$ and $\phi$ |
| $\boldsymbol{T_i}$ | Trajectory belonging to set $\boldsymbol{T}$ |
| $\boldsymbol{T}$ | Set of trajectories from a dataset |
| $\boldsymbol{T}_i$ | True trajectory of a vessel |
| $\boldsymbol{t}_{\boldsymbol{T}_i}$ | Time vector associated with $\boldsymbol{T}_i$ |
| $\boldsymbol{u}_t$ | DKF acceleration vectors at time step $t$ |
| $\boldsymbol{X}_c^k$ | A state belonging to the CN set |
| $\boldsymbol{X}_i$ | Predicted state by the SPNS |
| $\boldsymbol{X}_i$ | The SPNS vector containing the observed observations of a vessel |
| $\boldsymbol{x}_t$ | DKF state vector, or LRM position vector |
| $\boldsymbol{z}_t$ | DKF observed estimate |
| $\emptyset$ | Empty set |
| $\hat{\boldsymbol{x}}_t^-$ | LRM predicted position vector using all observations up until time-step $t-1$ |
| $\hat{\boldsymbol{x}}_0$ | DKF initial state vector |
| $\hat{\boldsymbol{X}}_i^{k+}$ | SPNS a posteriori state |
| $\hat{\boldsymbol{X}}_i^{k-}$ | SPNS *a priori* state |
| $\hat{\boldsymbol{x}}_{t-1}$ | LRM updated estimated position vector using all observation up until time-step $t-1$ |
| $\hat{\boldsymbol{x}}_{t-1}$ | Update state estimate |
| $\hat{\boldsymbol{x}}_t$ | DKF Updated state estimate |
| $\hat{\boldsymbol{x}}_t^-$ | Predicted state estimate |
| $\hat{\chi}^{k+}$ | SPNS *a posteriori* state COG |
| $\hat{\chi}^{k-}$ | SPNS *a priori* state COG |
| $\hat{\upsilon}^{k-}$ | SPNS *a priori* state SOG |

| | |
|---|---|
| $\mathcal{C}$ | Set of undersampled trajectories from $\boldsymbol{T}$ |
| $\mathcal{C}_{\boldsymbol{T_i}}$ | Under-sampled set from trajectory $\boldsymbol{T_i}$ |
| $C^k$ | CN set of the SPNS |

# CHAPTER 1

# INTRODUCTION

## 1.1 PROBLEM STATEMENT AND OBJECTIVES

The world's oceans are of critical importance to humanity as it is key to fisheries, shipping, and the environment. From an economic perspective, it is estimated that 90% of all global goods and energy transportation is done by sea, with millions of people being dependent on maritime-related activities for their livelihoods (Fang *et al.*, 2020*b*). As maritime activities increase globally, there is a greater dependency on technology for monitoring, controlling, and surveying of vessels and their activities. One of the most prominent systems for monitoring vessel activity is the Automatic Identification System (AIS).

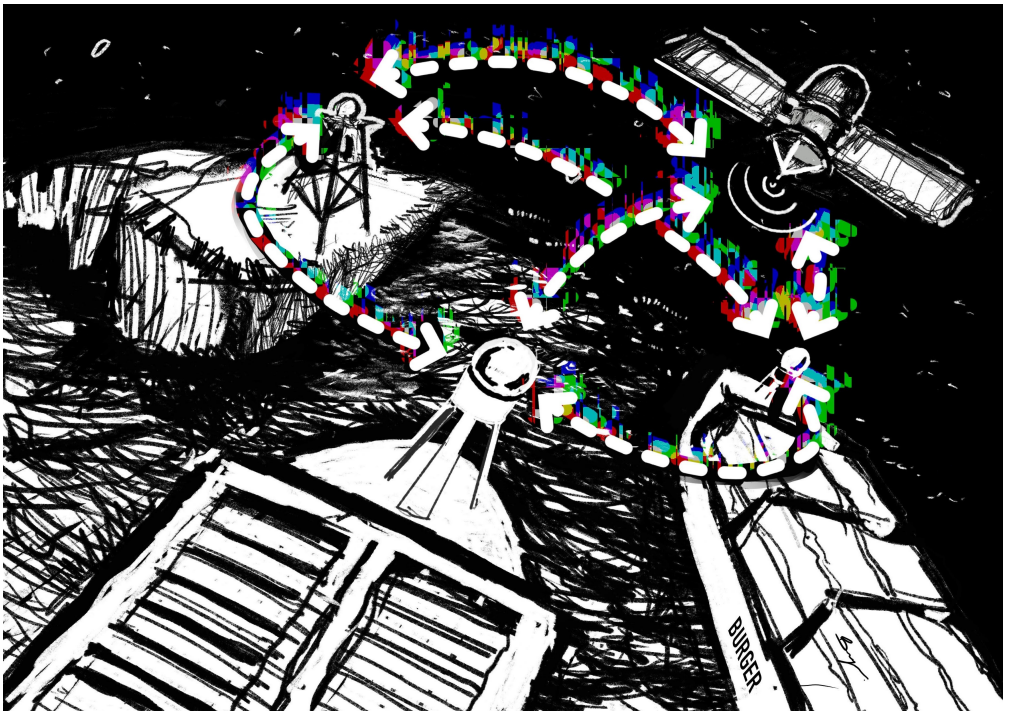


Figure 1.1: Artisanal depiction of the AIS system data flow (Burger, 2021)

AIS operates in the Very High Frequency (VHF) band and transmits messages from vessels to other vessels, terrestrial shore stations, and satellites (an artisanal depiction is shown in Figure 1.1). Due to the global increase in vessels fitted with AIS transmitters and the proliferation of satellite (S-AIS) and terrestrial (T-AIS) receiving stations, a significant increase in AIS messages

has been received globally. This increased data volume makes it possible to track the real-time movement of vessels and opens the door for improving vessel location predictions via historic vessel movement patterns. Several algorithms have been developed in recent years to aid in improved vessel coordinate prediction.

In this thesis, we present a novel prediction method whose main aim is to aid in vessel trajectory prediction of Cargo and Tanker vessels, specifically. We developed this method as we identified the need for a model with a low run-time complexity that is easy to implement and initialise. We will refer to this novel method as the Linear Regression Model with added *a priori* Course Information (LRMAC) throughout this thesis. When we designed the LRMAC, we kept in mind that it should be simple to implement and easy to initialise. LRMAC uses an Linear Regression Model (LRM) model at its core; and with the addition of *a priori* information whilst predicting, is able to predict non-linearly.

The *a priori* information the proposed method utilises is in the form of Spatial Maps (SMs). SMs are two-dimensional grids with cells containing information about specific geospatial locations on Earth.

**The main problem we address in this thesis is:**

Develop a low complexity, programmatically simple to implement method of predicting the trajectories of Cargo and Tanker vessels. Moreover, this simplistic approach should incorporate *a priori* information.

As we have already alluded to, the algorithm developed to solve the above problem is the LRMAC. Designing the LRMAC required multiple steps. First, a thorough literature review was conducted to investigate what approaches others have proposed to address the aforementioned problem. This literature review can be found in Section 2.2.2. Secondly, from this literature review, it seemed logical to use an LRM as our base model since Cargo and Tanker vessels travel along piece-wise linear trajectories. Next, it had to be determined if the LRM was an accurate and sufficient model to use for the prediction of Cargo and Tanker vessels' trajectories. The LRM and a more complex benchmarking approach, the so-called Discrete Kalman Filter (DKF), is presented in Section 4.3 and Section 4.2, respectively. The LRM and the DKF were

compared, and the result of this comparison study is discussed in Chapter 5.2.2. The LRM was found to be sufficient (to predict the linear segments of the piece-wise linear trajectories). The next step entailed extending the LRM to also incorporate *a priori* information, to allow the model to not only predict linear trajectories but also piece-wise linear trajectories. The problem then arose, what form should this *a priori* information take to enable such an extension? The format settled on were SMs, which are discussed in greater detail in Section 3.2. SMs summarise pertinent information, and its content could be extracted efficiently. Once the format was decided on, exactly how it should be incorporated to improve the prediction capability of the LRM needed to be established. The incorporation mechanism is discussed in greater detail in Section 4.5.2. Finally, we had to compare the accuracy and the time complexity of the developed method with a similar approach in literature. During the literature study, the Single Point Neighbour Search (SPNS) was identified as a good benchmarking approach. The SPNS is described in detail in Section 4.4. The result of this comparative study is presented in Section 5.3.3. In short, the outcome of this comparative study was that the LRMAC performed better than the SPNS in terms of long term prediction accuracy, and it has a significantly faster execution time.

## 1.2 RESEARCH QUESTIONS

The research questions of this thesis are:

1. How does an LRM compare (in terms of accuracy) to other more complex vessel trajectory prediction methods when we consider the Tanker and Cargo vessel use case?
2. Can the LRM be extended to predict long-term non-linear trajectories by incorporating *a priori* information?
3. How does this extended LRM compare with other similar *a priori* prediction methods from literature in terms of prediction capability and run-time complexity?

## 1.3 THESIS STRUCTURE

**Chapter 2** provides background on the vessel tracking technologies that currently exist. AIS is discussed, together with alternative and complementary technologies. Furthermore, the technolo-

gies are compared in detail, discussing their advantages and disadvantages. The hardware used by each technology is also discussed. We then further discuss AIS and methods that utilise AIS for the prediction of vessel coordinates, in the form of a literature study. The methods discussed range until mid-2021.

**Chapter 3** introduces the AIS dataset used to compare all the methods presented in this thesis. The cleaning of the dataset and its statistics are also discussed. We introduce SMs as well and how they were generated from the aforementioned dataset.

**Chapter 4** discusses all the methods included in this study. We start with the non *a priori* methods[1], the DKF and LRM and then introduce the *a priori* methods the SPNS and the created novel LRMAC.

**Chapter 5** contains the results of the method comparisons and the experimental design for each of them. An in-depth comparison for each of the non *a priori* and *a priori* methods are presented, where all the methods were tested on multiple vessels. Each comparison is followed by a case study.

**Chapter 6** concludes the thesis, the specific drawbacks and advantages of the various methods are discussed, highlighting some of the observations that can be made from this work.

**Appendix A** provides an in-depth overview of the DKF. The appendix discusses the DKF in detail and should equip the reader to have a good understanding of the DKF and how it works. An example of the DKF example is also presented and discussed.

**Appendix B** provides an in-depth overview of the LRM. The least squares (LS) fit is discussed in detail. An example is presented and discussed to allow the reader to understand how the LS fit works and how the optimal LRM fit is obtained.

**Appendix C** introduces two tables. These tables contain the information of the vessels and trajectories used to test the non *a priori* and *a priori* methods on.

[1]When we refer to non *a priori* methods throughout this thesis, we refer to methods that do not continuously incorporate historic information (*a priori*) whilst predicting. We are aware that the DKF makes use of *a priori* information during the initialisation of its matrices, and it also refers to the previous predicted state as the *a priori* state estimate.

## 1.4 ACADEMIC CONTRIBUTIONS

In this section we highlight the academic contributions made during the course of the study.[2]

### 1.4.1 Conference Papers

- C. N. Burger, T. L. Grobler and W. Kleynhans, "Discrete Kalman Filter and Linear Regression Comparison for Vessel Coordinate Prediction", 2020 21st IEEE International Conference on Mobile Data Management (MDM), 2020, pp. 269-274.
- T. L. Grobler, W. Kleynhans, B. P. Salmon and C. N. Burger, "Unsupervised Sequential Classification of Modis Time-Series", IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 2244-2247.

### 1.4.2 Journal Papers

- C. N. Burger, W. Kleynhans, and T. L. Grobler, (2021)., "Extended Linear Regression Model for Vessel Trajectory Prediction with *a priori* AIS Information", *Geo-spatial Information Science Journal*, *In Press*.

### 1.4.3 GitHub

All code used throughout this thesis is documented and available on a public GitHub repository. The repository contains relevant code and algorithms, and the link is provided below:

- `https://github.com/cnburger/MScComputerScience`

[2]The content presented in this Thesis supersedes and improves any work published before the year 2021.

# CHAPTER 2

# LITERATURE REVIEW

In this chapter we introduce different vessel tracking technologies and techniques considered. We discuss each technology and do a comparison between them. Furthermore, we do a deeper dive into one of the vessel tracking technologies, namely, AIS. During the deep dive we discuss AIS applications followed by a literature review on prediction methods that make use of AIS.

## 2.1 VESSEL TRACKING TECHNOLOGIES

In recent years the tracking and monitoring of vessels have become more prevalent and necessary to improve the safety of maritime-related activities and the crews involved.

At the United Nations (UN) Geneva Convention of March 1948, a convention adopted at the UN Maritime Conference resulted in the formation of the International Maritime Organisation (IMO), which was formally known as the Intergovernmental Maritime Consultative Organisation (IMCO) until 1982. In 1948, the convention was prepared and opened for signature, and its acceptance was convened by the Secretary-General of the UN (United Nations, 1958). The IMO officially came into force in March 1958. In short, the Geneva Conventions and their additional protocols form the core of international humanitarian law, which regulates the conduct of armed conflicts and seeks to limit the effects thereof[1].

The International Convention for the Safety of Life at Sea (SOLAS), is an international maritime treaty that sets the minimum safety standards for the operation of vessels, created by the IMO (United Nations, 1980). The increase in tracking and monitoring of vessels is due to the SOLAS regulations created by the IMO, and the worldwide adoption thereof. These regulations require vessels that meet certain criteria to be fitted with specific transponders, which aid in vessel monitoring.

A transponder is a device capable of automatically transmitting and receiving signals (data). Various types of transponder technologies to monitor vessel activity exists. One such technology is AIS, which is an active tracking technology consisting of both terrestrial-AIS (T-AIS) and satellite-AIS (S-AIS). The aforementioned transmitters transmit data at regular time intervals (Curlander and

[1]For more on the Geneva Convention see: `https://www.icrc.org/en/war-and-law/treaties-customary-law/geneva-conventions`

McDonough, 1991).

AIS is one of the most common tracking technologies used by vessels. However, as we have already alluded to, there exist a wide range of different technologies similar to AIS with different use cases and applications. One such technology is Long Range Identification and Tracking (LRIT). LRIT is a satellite-based system that samples at a lower rate than AIS. An advanced non transponder based, tracking technology called Synthetic Aperture Radar (SAR) also exists, which is used as a complementary technology to AIS and LRIT. SAR is a satellite/aircraft-based system able to track vessels in all terrains and weather conditions. Optical satellites and coastal radar are also complementary technologies to AIS and LRIT, allowing for improved tracking of vessels.

A shore-side monitoring service exists that utilises all the different tracking technologies, known as the Vessel Tracking Service (VTS). The VTS utilises technologies, such as radar, AIS, LRIT and other visual aids.

### 2.1.1 Vessel Tracking Service

The VTS is a marine traffic monitoring system created by harbour and port authorities to monitor vessel activity close to shore. The first VTS system appeared at a port in Liverpool, UK in 1949[2]. The VTS can contact vessels from shore using radio frequency transmission; relaying important information identified by all of the technologies in the VTS control room onshore, which human operators manage. The VTS is similar to air traffic control for aircraft. The type of information communicated to vessels from the VTS includes: positions of other traffic, meteorological hazard warnings such as disturbances on the sea floor and bathymetric[2] information. The VTS only manages traffic within a port or waterway that is within its range of authority.

The VTS has predefined zones of authority enabling captains of vessels to make more informed navigational decisions. The ocean can be divided into these specific areas (zones). The zones are visualised in Figure 2.1 and summarised below.

1. Region A1, an area which lies within the coverage region at least one Very High Frequency (VHF) coast station providing Digital Selective Calling (DSC) alerting and radiotelephony services.

[2]According to the United States Coast Guard, see: `https://www.navcen.uscg.gov/?pageName=vtsHistory`
[2]*Bathymetry* is defined as the measurement of the depth of oceans, rivers, or lakes.

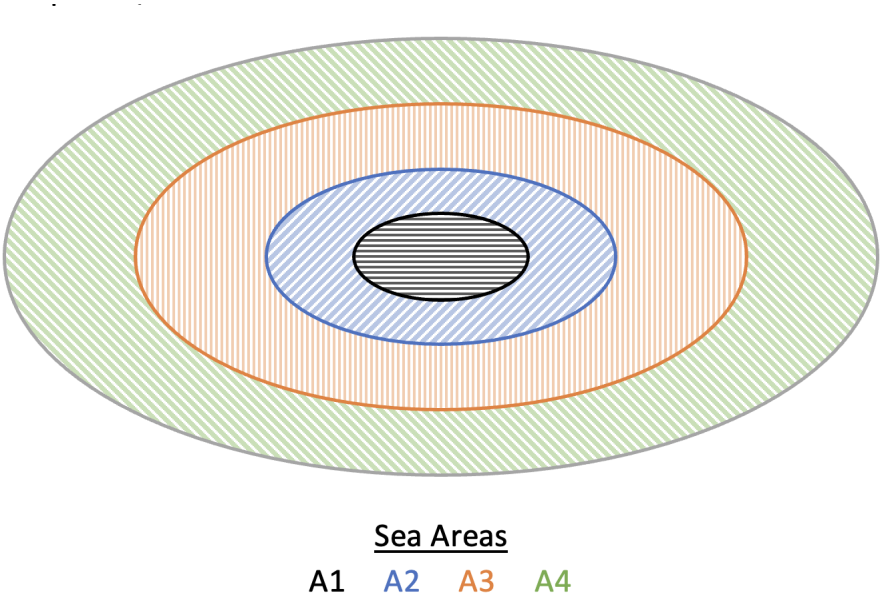


Figure 2.1: VTS visualisation of sea area zones

2. Region A2, an area excluding A1, which is within the radiotelephone range of at least one Medium Frequency (MF) coast station in which continuous DSC alerting and radiotelephony services are available.

3. Region A3, an area excluding A1 and A2, within the coverage of an Inmarsat geostationary satellite in which continuous alerting is available.

4. Region A4, an area outside A1, A2 and A3. Vessels must carry a DSC-equipped High Frequency (HF) radiotelephone/telex.

The VTS's authority is limited to a specific area within a certain radius of the control centre. When a vessel enters a VTS area, they must report to the authorities using radio. Vessels may be tracked by the VTS control centre as determined by the IMO. When vessels are within a VTS area, they must monitor a specific radio frequency for warnings, and they can be contacted directly by a VTS operator in the event of an emergency, risk of incident or where traffic flow is regulated by the VTS. SOLAS regulation V/12 states that governments may establish a VTS when in their opinion, the volume of traffic or the degree of risk justifies such services. The primary purpose of the VTS is to improve the safety and efficiency of harbours, ports and routes.

VTS utilises various transponder and non-transponder based technologies. Non-transponder based

technologies include radio, radar, optical satellites, and SAR. Transponder based technologies include AIS and LRIT. Radar can be used by the VTS to detect the size of an object and its speed. Vessels equipped with radar systems can further extend the range of radar, transmitting the collected information to the VTS. AIS aid in the reliability and efficiency of navigation, utilising terrestrial receivers (T-AIS) and satellites (S-AIS) to transmit data. VTS makes use of a technology known as Radio Direction Finder (RDF) which assists in collecting maritime information and locating the direction from which radio frequencies are coming.

Vessels have to carry specific transponders or a range of transponders depending on the criteria set by the IMO. There are different types of transponders that can be employed, namely AIS and LRIT.

### 2.1.2 Automatic Identification System

The AIS system was developed to identify, track, and report on different types of data points generated by vessels. Data availability and coverage of AIS has increased significantly over time. The usage of AIS for improved vessel tracking, location predicting, position monitoring and collision avoidance has also grown significantly (Yang *et al.*, 2019). Large volumes of AIS data are being recorded each day, Natale *et al.* (2015) reported that in 2014 there were 200 million unique AIS messages that were recorded in each month, amounting to 6.5 million AIS messages a day. Yang *et al.* (2019) mentioned that if an AIS message is sent every 10 seconds from a single vessel, it will send up to 3 million messages a year. AIS messages can be categorised as static messages, dynamic messages, voyage related messages and safety-related messages (Harchowdhury *et al.*, 2012). A subset of the information AIS messages contain is shown in Table 2.1.

AIS information is broadcasted, collected and exchanged on a regular basis (Balduzzi *et al.*, 2014). The frequency thereof varies from a couple of seconds to minutes depending on the type of information being sent and the condition of a station (some AIS receiver technologies are old, and a delay in signal processing may occur). All AIS messages have an identifier that indicates which vessel the message belongs to, called the Maritime Mobile Service Identity (MMSI). A MMSI is issued by authorities such as the United States coast guard, which also issues the call sign of a vessel

[3]World Geodetic System (WGS-84) is a language of location and it is used by the Global Positioning System (GPS) (Kumar, 1988)

| Description | Measurement Unit | Range |
|---|---|---|
| Maritime Mobile Service Identity (MMSI) | - | 9 Digits |
| AIS Navigational Status | Integer | 0-100 |
| Longitude (LON) | Degrees (WGS-84[3]) | [-180.0000, 180.0000] |
| Latitude (LAT) | Degrees (WGS-84) | [-90.0000, 90.0000] |
| Speed over Ground (SOG) | knots (kt) | [0.0, 102.0] |
| Course over Ground (COG) | Degrees | [0.0, 359.0] |
| Rate of Turn (ROT) | Degrees per min | [0, 720] |
| Heading | Degrees | [0, 359] |
| Bearing | Degrees | [0, 359] |
| Timestamp (Coordinated Universal Time) | UTC | - |

Table 2.1: Information in an AIS message

(Balduzzi *et al.*, 2014). The MMSI consists of numbers whereas the call sign is an alphanumeric string.

The MMSI consists of nine digits, and uniquely identifies each vessel. A MMSI can be used on different types of vessels and crafts. The MMSI encoding for vessels are indicated in Table 2.2 ( Saputra *et al.* (2018)). The first three digits of the MMSI refer to the Maritime Identification Digits (MID) which specifies the administration (country) or geographical administration responsible for the vessel, as stipulated by the International Telecommunication Union (2012) recommendation M.585-6. The subscript in Table 2.2 refers to the position number in the MMSI string, the prefix values [0, 1, 9] are used to identify a certain type of vessel/craft, the characters [$M$, $I$, $D$] represent the three digit MID integer values, and $X$ represents any integer values from zero to nine, inclusive.

| Vessel/Craft Type | MMSI String Structure |
|---|---|
| Individual Vessel | $M_1\ I_2\ D_3\ X_4\ X_5\ X_6\ X_7\ X_8\ X_9$ |
| Group of Vessels | $0_1\ M_2\ I_3\ D_4\ X_5\ X_6\ X_7\ X_8\ X_9$ |
| Shore Station | $0_1\ 0_2\ M_3\ I_4\ D_5\ X_6\ X_7\ X_8\ X_9$ |
| SAR Aircraft | $1_1\ 1_2\ 1_3\ M_4\ I_5\ D_6\ X_7\ X_8\ X_9$ |
| Navigation Aids | $9_1\ 9_2\ M_3\ I_4\ D_5\ X_6\ X_7\ X_8\ X_9$ |

Table 2.2: MMSI encoding breakdown

The IMO adopted the regulation from SOLAS which requires that certain types of vessels must be fitted with an AIS transmitter (the regulations are listed below). SOLAS Chapter V Regulation 19/.2.1.4 paragraph 2.10, which specifies which vessels engaged on international voyages ought to be fitted with an Electronic Chart Display and Information System (ECDIS), to improve the safety

of vessels and their crews at sea. An AIS transmitter is an ECDIS.

The aforementioned regulations as they pertain to Cargo and Tanker vessels are summarised below:

- Tanker Vessels
  - 3000 Gross Tonnage (GT) and upwards constructed on or after 1 July 2012.
  - 3000 GT and upwards constructed before 1 July 2012; no later than the first survey[4] on or after 1 July 2015.
- Cargo Vessels
  - 10000 GT and upwards constructed on or after 1 July 2013.
  - 50000 GT and upwards constructed before 1 July 2013; no later than the first survey on or after 1 July 2016.
  - 20000 GT and upwards but less than 50000 GT constructed before 1 July 2013, with the first survey on or after 1 July 2016.
  - 10000 GT and upwards but less than 20000 GT constructed before 1 July 2013; no later than the first survey on or after 1 July 2018.

#### *2.1.2.1 AIS Transponders*

AIS transponders make use of VHF radio and GPS technology. As mentioned the AIS communication medium can be broken into two types which are used interchangeably, namely T-AIS and S-AIS. Transponders can transmit to onshore receiving stations, other AIS equipped vessels in the vicinity and satellites. AIS transmission range is similar to that of VHF radios. AIS broadcasts information in packets, where each packet contains 256 bits transmitted at 9600 bits per second (bps) (Harchowdhury *et al.*, 2012). In Figure 2.2, a simplistic artisanal depiction of the AIS transponder communication network is shown (created specifically for this thesis).

[4]The *first survey*, is defined in a regulation of the 1974 SOLAS convention. The first survey refers to the "first annual, periodical or renewal survey, whichever is due first after the date specified in the relevant regulation or any other survey if the Administration deems it reasonable and practicable." `https://www.imorules.com/GUID-A0F84557-EF59-4E18-BB75-34C1FC2F0DF7.html`.

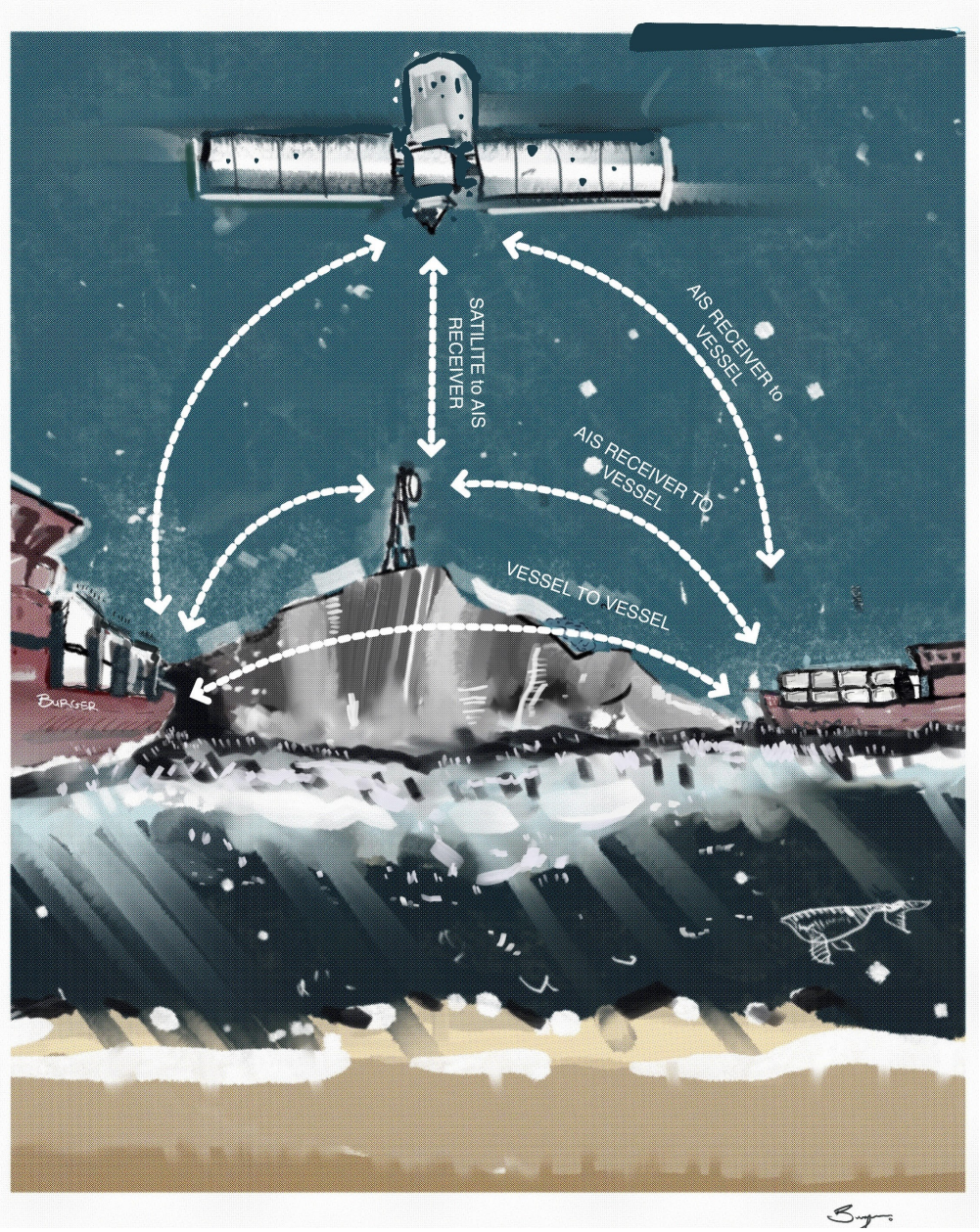


Figure 2.2: An artisanal depiction of the AIS transponder communication network, S-AIS and T-AIS combined (Burger, 2021))

AIS transponders can be divided into two classes, A and B. Class A transponders have stricter requirements compared to class B transponders. Class A transponders are required on vessels in excess of 300 GT that partake in international voyages, where class B can be fitted to any non-SOLAS vessel. Class B transponders have a lower cost compared to class A transponders and are much simpler. Class A transponders comply with all relevant IMO AIS requirements. In Table 2.3 a comparison between Class A and B transponders is provided.

| Feature | Class A | Class B |
|---|---|---|
| **Transmit power** | 12.5 Watts (W)<br>2 W (low power) | 2W |
| **Unique Communication Access Scheme** | Self Organised Time Division Multiple Access [5] (SOTDMA) | Carrier Sense Time-Division Multiple Access[6] (CSTDMA) |
| **Transmission Frequency Range** | 156.025-162.025 MHz @12.5/25 kHz<br>DSC (156.525 MHz) | 156.025-162.025 MHz @25 kHz<br>Optional: DSC (156.525 MHz) & 12.5kHz |
| **Miscellaneous** | External GPS<br>Heading<br>Rate of Turn Indicator | Heading (optional) |
| **Safety Text Messaging** | Transmit and Receives | Transmit (optional) |

Table 2.3: Class A and B, AIS transmitter information

#### *2.1.2.2 AIS drawbacks, vulnerabilities and security risks*

Balduzzi *et al.* (2014) did an in-depth evaluation of AIS regarding security. Like any system, AIS also has its vulnerabilities and is constantly being improved to be more resistant against cyber-attacks. Androjna *et al.* (2021) did an in depth investigation into AIS vulnerabilities and how these vulnerabilities can and are being exploited. These vulnerabilities have been reported to the relevant organisations.

[6]SOTDMA is an access protocol used by AIS for continuous transmission. The idea behind SOTDMA is that vessels within a self organised area maintain timing synchronisation to transmit AIS messages amongst themselves autonomously (Harchowdhury *et al.*, 2012). Each SOTDMA area can be thought of a circle with a radius of 20 Nautical Miles (NM) from the AIS equipped vessel. Vessels will transmit information to other vessels within its SOTDMA area.

[6]CSTDMA is an access protocol to provide channel access and simultaneous switching. Carrier sensing is used to defer transmissions until no other stations are transmitting (Zhang *et al.*, 2014). Once a free slot is found, transmission of information starts, for more see and article by Weatherdock: `https://www.easyais.com/en/technical-news/ais-know-how-data-transfer-sotdma-vs-cstdma/`

One of the drawbacks of AIS is that vessels may deactivate their AIS transponders. According to the criteria laid out by SOLAS and adopted by the IMO, not all vessels have to be equipped with AIS transponders; only certain vessels have to be equipped with a transponder, namely those that fulfil the criteria set by SOLAS (Chen *et al.*, 2019).

The three main threats of AIS identified by Balduzzi *et al.* (2014) are spoofing, hijacking and availability disruption. These threats can be software or radio frequency related, disrupting their normal function. The malicious intent behind these attacks is to cause dismay in: vessel tracking, identification, prediction, and monitoring.

AIS spoofing includes the creation of artificial vessels that interfere with Search and Rescue Transponders (SARTs) where messages are sent to. These messages are usually in the form of fraudulent emergency messages, luring a targeted vessel to a hostile and attacker-controlled sea space (Balduzzi *et al.*, 2014). By law, a vessel must join a search and rescue sea operation.

AIS hijacking happens when any information in AIS messages is altered, compromising the integrity and quality of the data collected from AIS equipped vessels. Information sent to vessels can be modified maliciously, letting vessels veer off course. In both cases, the recipients receive attacker modified versions of the original AIS messages (Balduzzi *et al.*, 2014).

Finally Balduzzi *et al.* (2014), also mentioned the existence of availability disruption attacks. These attacks' purpose is to interfere with radio frequencies. Attacks include messages being sent to transponders, instructing them to change their broadcasting frequencies, rendering AIS useless. When the transmission frequencies change, the data sent will not be recorded by the AIS receivers. The interruption of communications within a certain coverage range can result in large scale AIS disruptions. Another type of disruption attack is the so-called timing attack. A timing attack causes AIS transponders to delay transmission times and prevent them from communicating their position and other relevant tracking information.

Although vulnerabilities do exist within the AIS system, the system is reviewed, maintained and updated more regularly than any other vessel tracking system (Balduzzi *et al.*, 2014).

### 2.1.3 Long-range Identification and Tracking (LRIT)

LRIT is the most similar to AIS if compared to all the vessel tracking technologies, although it has a completely different data pipeline and use case. The LRIT system allows for the global identification and tracking of vessels to improve the security of shipping, safety, and maritime environments. According to the European Maritime Safety Agency (EMSA), LRIT was established in October 2006 by the IMO[7].

LRIT is a satellite-based real-time reporting system that collects and distributes vessel positional information to specific LRIT data centres (DC) as determined by each vessel's Flag administration (Xiao *et al.*, 2020). A Flag administration can be defined as the country/nation a vessel belongs to. Each Flag administration has a list of vessels entitled to fly its flag, which means that the associated LRIT data is governed by the rules set out by the Flag administration. Each administration determines the DC to which LRIT information has to be sent to, from the vessels with their flag.

The majority of the Flag administrations opted for cooperative DCs. However, there are nations that opted for National DCs, such as the US where DC data is not shared unless otherwise specified. Each Flag administration has the right to protect the information of their vessels, while allowing Coastal states to access information of a vessel that is about to enter their coastal region (Verma, 2009). A coastal state is defined as any nation with a coast. Coastal states have the right to protect their coastline as set out in the United Nations Convention on the Law of the Sea (UNCLOS) created by the United Nations General Assembly (1982). The US, as an example, has full control over their LRIT data (national DC), and has to provide their LRIT data to coastal administrations when vessels under their flag enters their coastal region as stated in UNCLOS and SOLAS. Every Flag administration is responsible for enforcing any applicable maritime regulations. LRIT entitles all SOLAS Contracting Governments to receive information about vessels, that is up to 1000 NM from their coast (Verma, 2009).

[7]The EMSA is an agency of the European Union, more on LRIT from the EMSA is available at `http://www.emsa.europa.eu/lrit-main/lrit-home/legal-basis.html`

The LRIT system as a whole consists of several items. The components of an LRIT system are listed below (International Maritime Organization, 2020):

- LRIT information transmitting and satellite communication equipment
- Communication Service Providers (CPS)
- Application Service Providers (ASP)
- LRIT DC, including any related Vessel Monitoring System (VMS)
- LRIT Data Distribution Plan (DDP)
- LRIT International Data Exchange (IDE)

The LRIT system by default should transmit the positional information of a vessel in no less than six-hour intervals to the respective DCs (There should be at least four positional updates in a 24-hour period). The transmission frequency is determined by the Flag administration. The types of information transmitted by LRIT can be seen in Table 2.4 below.

| Description | Measurement | Range |
|---|---|---|
| LON WGS-84 | Degrees | [-180.0000°, 180. 0000°] |
| LAT WGS-84 | Degrees | [-90.0000, 90.000] |
| Timestamp (UCT) | UCT | |
| Ship name | String | |
| IMO Number | | |
| Call Sign | | |
| Martime Mobile Service Identity (MMSI) | | 9 Digit Number |

Table 2.4: LRIT Tracking Parameters

The LRIT regulation applies to all vessel types on international voyages, as required and stipulated by the International Maritime Organization (2020). Vessels that needs to be equipped with LRIT are listed below:

- All passenger vessels including high-speed craft.
- Cargo vessels, including high-speed craft of 300 gross tonnage (GT) and above.
- Mobile offshore drilling units.

### 2.1.4 Complementary Technologies

In this section, we discuss complementary technologies to AIS and LRIT. These technologies include SAR, Coastal based Radar, and Optical satellites, all non-transponder based. They are used to further improve transponder-based systems' tracking abilities.

#### *2.1.4.1 Synthetic Aperture Radar*

Synthetic Aperture Radar (SAR) is a satellite/aircraft-borne technology, that can be utilised for maritime surveillance. SAR's biggest advantage is that it can operate in any weather condition and time of day. SAR imagery can track vessels at sea, as they are highly reflective when compared to the oceanic background (Schwegmann *et al.*, 2018). SAR, however, cannot provide further information on a vessel other than its location and dimensions, therefore it is being used together with transponder technologies.

The physical SAR antenna is relatively small and can provide high-resolution large-scale imagery (Curlander and McDonough, 1991). However, SAR simulates the use of a long antenna by the implementation of signal processing techniques (Cutrona, 1990), enabling higher resolutions given the small antenna size[8]. SAR uses the fact that it is fitted to a moving object to its advantage, producing higher resolution images through the means of signal processing. The reduced size of the SAR system means the incurred cost to implement is also reduced, as the size and weight of a satellite-borne system have a significant impact on the cost thereof (Chan and Koo, 2008).

SAR imagery only covers areas where satellites with the technology are actively at (due to satellite orbits). AIS technology, on the other hand, is not only dependent on satellites but has terrestrial receivers as well.

If the right operating frequency is chosen for SAR, it will be able to penetrate through clouds, rain and fog without any diminishing effects on the quality of SAR imagery. Therefore, SAR can be used in any weather condition, while optical and infrared systems are not able to operate in all weather conditions (Ulaby *et al.*, 1981). SAR is an active sensor, which means that it has its own source of illumination (not relying on sunlight reflecting off of the earth); therefore, it can operate at any time of day (Chan and Koo, 2008).

[8]Longer antennas usually translates into higher spatial resolutions imagery, and improved data capturing

SAR was developed to overcome coverage limitations of traditional technologies, which require vessels to have transmitting equipment on board. A SAR equipped aircraft can be deployed in any region to identify and track vessels. Schwegmann *et al.* (2017), made use of SAR to accurately detect vessels at sea. Multiple other methods that utilise SAR have also been developed. Other examples include Koppe *et al.* (2014), Brusch *et al.* (2010) and Wang *et al.* (2017).

SAR has various applications, and is not only limited to vessels at sea (Chan and Koo, 2008). Other applications include:

- Mining (Lynne and Taylor, 1986)
- Oceanography (Walker *et al.*, 1996) such as Bathymetry (Ma *et al.*, 2021)
- Oil pollution and Environment Monitoring (Hovland *et al.*, 1994)
- Sea ice (Drinkwater *et al.*, 1990) and snow monitoring (Storvold *et al.*, 2006)
- Terrain Classification (Kong *et al.*, 1990)
- Vessel monitoring and surveillance

SAR and AIS have been used together in the past for maritime surveillance. Achiri *et al.* (2018), proposed a method to fuse SAR images and AIS data for improved vessel detection and feature extraction. Various SAR and AIS algorithms exist. See Zhao *et al.* (2014) for a review on such algorithms.

SAR, however, does not allow for continuous temporal coverage of a specific region together with adequate real-time surveillance (Maresca *et al.*, 2014). SAR requires complex signal processing to make use of its data (Chaturvedi, 2019). Raw SAR data has to be transformed into usable data, whereas AIS data needs no transformation from its raw state. AIS transmitters already transform the measurements it makes into usable data; in the form of AIS messages.

#### *2.1.4.2 Coastal Radar System*

The Coastal Radar System (CRS) consists of a range of onshore radar stations along the coast. Coastal Radar (CR) allows for the detection of vessels within the range of operation defined by the Radar System (RS). The range of a RS is dependent on the technology and specifications of the system. CRS, similar to SAR, is used in conjunction with transponder based technologies to

further improve the tracking of vessels.

RS play an important role in the monitoring of sea and air traffic; reducing accidents in the transportation sector (Octavian and Jatmiko, 2020). The RS utilises electromagnetic waves for the detection of objects. It can measure distances from the radar to any object, making maps of objects within its range (these objects includes planes, vessels and vehicles). The RS is also capable of detecting weather information and including it on its maps. Octavian and Jatmiko (2020), proposed a method to aid in MSA by using AIS, CRS and long-range cameras together with Artificial Intelligence (AI) to improve vessel tracking, anomaly detection and path planning.

High Frequency Radar (HFR), can have a detection range of up to 200 km. However, detection accuracy is limited due to interference by external factors (see Dzvonkovskaya and Rohling (2010)). Conventional microwave radars operate in line-of-sight propagation, and as such, are limited to a few kilometres (Maresca *et al.*, 2014). CRS is widely used and is the most common technology which is used by the VTS.

#### *2.1.4.3 Optical Satellites*

Optical satellite tracking involves the classification and identification of vessels by means of optical images captured from satellites, such as Sentinel-2[9]. Optical refers to the visible spectrum of the human eye, with wavelengths ranging from $400 - 700$ nanometre (nm), and reflected infrared that covers the near and short-wave infrared bands of up to $3\mu$m (Kanjir *et al.*, 2018). Optical sensors are passive sensors, which means that they rely on an external illumination source, such as the sun.

Optical imagery plays an important role in maritime surveillance; literature studies on detection algorithms utilising optical imagery have been published as early as 1978 (Kanjir *et al.*, 2018). Kanjir (2019) proposed a method using freely available Sentinel-2 optical image data to support humanitarian efforts to identify migrants that risk their lives to migrate over the Mediterranean sea, illegally on makeshift vessels. Vessel detection is also essential in sea rescue operations, or to detect illegal activity such as pollution, illegal fishing, and illegal migration. Kanjir *et al.* (2018), did an in-depth review of the literature that makes use of optical satellite imagery for vessel detection and

[9]Sentinel-2 is an earth observation mission launched by the joint effort between the EC (European Commission) and ESA (European Space Agency). Sentinel-2 is a constellation of two identical satellites (Sentinel-A and Sentinel-B) in the same orbit, phased at 180° relative to each other (`https://sentinel.esa.int/web/sentinel/missions/sentinel-2`).

classification, up to 119 studies were reviewed.

### 2.1.5 Technology Comparison

AIS and LRIT can be used to track and identify vessels whereas SAR, Optical Satellites and CRS are usually used together with AIS and LRIT, further improving the tracking abilities. Some technologies are superior to others and have numerous advantages, but it comes with a trade-off which is usually cost-related.

CR is the most limited of all the technologies as it can only track vessels close to shore and in waterways, whereas the other technologies have global coverage due to satellite technology. When comparing AIS with Optical Satellites, it is far superior, as it allows for vessel tracking in any weather condition and time of day. SAR technology comes at a relatively high cost. LRIT data is exclusive and limited to the LRIT network. AIS makes use of a combination of terrestrial and satellite data, allowing for worldwide coverage, and it tracks multiple parameters of a vessel at a relatively low cost. The process to obtain AIS data is relatively easy.

Although each system has its drawbacks and compromises, hybrid systems exist. Hybrid systems aid in the improvement in vessel tracking and classification abilities leading to more accurate results, i.e.:

- Chaturvedi *et al.* (2012) made use of a hybrid SAR and AIS method for improved vessel trajectory prediction.
- Lang *et al.* (2018) used AIS together with SAR to improve vessel classification.
- Milios *et al.* (2019) combined AIS and SAR for improved vessel classification. Kleynhans *et al.* (2013) made use of SAR and LRIT data for vessel detection.
- SAR can be used in conjunction with optical satellite imagery to produce images with even higher spatial resolutions (Xiao *et al.*, 2020).
- Meraner *et al.* (2020) proposed using SAR together with Optical Satellites for cloud cover removal from images, as optical imaging sometimes requires extra processing for the removal thereof.

In Table 2.5 each of the technologies are given and compared showcasing their advantages and disadvantages.

| | AIS | LRIT | Coastal Radar | SAR | Optical Satellites |
|---|---|---|---|---|---|
| **Detection Mechanism** | Self-reporting | Self-reporting | Proactive detection | Proactive detection | Proactive detection |
| **Primary Surveillance Area** | Ports and open oceans | Within 1000 nautical miles of coast | Port waters and EEZ[10] | Open Oceans | Open Oceans |
| **Update/ Recording frequency** | **Moving**: 2-10 seconds **Anchor**: 3 minute (Class-A AIS) | At least: 6 hourly and 4 updates a day | Configurable under the system limit | Configurable under the system limit | Configurable under the system limit |
| **Land/Air Based** | Shore and Satellite based | Satellite based | Shore based | Satellite/aircraft based | Satellite based |
| **Data format** | Decoded as value-based data | Decoded as value-based data | Radar Imagery | SAR imagery | Optical Imagry |
| **Data Cost and availability** | Readily available and low cost | Data exclusive | Limited to radar stations | High-cost data | High-cost data |

Table 2.5: Vessel Tracking Technology Comparison

## 2.2 AUTOMATIC IDENTIFICATION SYSTEM: A DEEP DIVE

AIS data has opened up new research possibilities into the behaviour of vessel movements. Research related to vessels equipped with AIS is still growing, with new studies and applications being published each year. AIS data is used by all the methods implemented in this thesis. In this section, we briefly discuss the applications of AIS allowing the reader to have an overview of the different AIS applications, followed by an AIS trajectory prediction literature review.

[10]Exclusive Economic Zones (EEZ) were determined by United Nations General Assembly (1982). It states that any coastal state can assume jurisdiction over the exploration and exploitation of marine resources in its coastal waters. Coastal waters of a state consist of the area within 200 NM of the state's shore lines.

### 2.2.1 Applications

One of the main use cases for AIS is Maritime Situational Awareness (MSA). Volumes of AIS data are being recorded each day, and the availability thereof has led to an increase in research of vessels and their behavioural patterns. AIS is used to gain more insight into vessel behavioural patterns (Hart and Timmis, 2008).

Data mining, in terms of AIS, is the process of extracting valuable information out of AIS data. Data mining can be divided into different categories: pattern recognition ((Xiao *et al.*, 2020) (Pallotta *et al.*, 2014)), concept-learning, prediction, clustering (Theodoropoulos *et al.*, 2019), and classification (both supervised and unsupervised) (Pitsikalis *et al.*, 2021) (Lang *et al.*, 2018). Trajectory extraction is another important process often applied to AIS data that entails the reconstruction or prediction of vessel trajectories (Yuan *et al.*, 2019).

Furthermore, anomaly detection, in terms of AIS, is the process of identifying vessels that do not conform to the normal and known behaviour typically observed by a vessel (Ristic *et al.*, 2008). Anomaly detection is often used to identify vessels with abnormal behaviour (Rong *et al.*, 2020). Anomaly detection is important due to the sheer volumes of vessel data,making it rather difficult for human operators at the VTS to monitor all vessels simultaneously.

### 2.2.2 Literature Review: Prediction

In the previous section, we briefly mentioned some applications of AIS. Recall that the main use case we focus on in this thesis is prediction.

As such, we present a literature review on different trajectory prediction methods that utilises AIS data in this section. Methods range from simplistic linear regression Machine Learning (ML) models, as done by Burger *et al.* (2020), to more advanced ML models such as that proposed by Xu (2020). As part of this thesis, we present a novel trajectory prediction method. Our literature review spans published methods until mid-2021 and will be discussed in chronological order.

Perera *et al.* (2012) proposed an Extended Kalman Filter (EKF) to estimate the state of a vessel. The EKF was also used for the prediction of vessel trajectories. The EKF as-is, can be used to fuse non-linear system kinematics with a set of noisy measurements. An Artificial Neural Network (ANN) is also introduced as a mechanism to detect and track multiple vessels. It is well known that

ANNs can self adapt, approximate universal functions, capture non-linear behaviour and compute posterior probabilities. The proposed EKF estimates the vessel's state, i.e. its position (spatial location), velocity, and acceleration. The acceleration is estimated by using position measurements that the ANN provides. A curvilinear motion model is selected so that the motions of vessels affected by external disturbances can be represented by white Gaussian noise. Perera *et al.* (2012) showed that the EKF could successfully predict future states and trajectories with acceptable error rates. The methods were tested on simulated examples with a time horizon of 50 seconds.

Pallotta *et al.* (2014) presented a vessel trajectory prediction method based on Ornstein-Uhlenbeck (OU) stochastic processes, where the parameters of these processes are estimated from historic AIS data. The data is clustered into three types: vessels, waypoints, and routes. Route extraction is done using Traffic Route Extraction and Anomaly Detection (THREAD) from AIS data, which is presented in Pallotta *et al.* (2013). The three types of clustered data aid in, vessel prediction and empirical calculations. The maximum prediction time window of a vessel depends on the mean duration of the historically observed route.

Mazzarella *et al.* (2015*a*) proposed a Bayesian vessel prediction algorithm based on a Particle Filter (PF). The PF (also known as Sequential Monte Carlo) is defined as a numerical approximation of the non-linear Bayesian filtering problem, used to filter and smooth state-space models (Gordon *et al.*, 1993). The proposed method refines the strategy presented by Mazzarella *et al.* (2015*b*) by exploiting algorithms in the field of Bayesian non-linear filtering. Mazzarella *et al.* (2015*a*) also proposed an architecture to fuse historic AIS, LRIT and VMS data (the results obtained by Mazzarella *et al.* (2015*a*) were all based on the fused data). The aforementioned fused data allowed for improved knowledge extraction from traffic patterns in the regions of interest. A Knowledge Based Particle Filter (KB-PF) approach is proposed, which was inspired by Papi *et al.* (2012) and Ristic *et al.* (2008) and is able to predict vessel motion patterns (i.e. the vessel's COG and SOG). The KB-PF was compared to a Knowledge Based Velocity Model (KB-VM) which was presented in Mazzarella *et al.* (2015*b*). The KB-VM is a more computationally efficient model; however, the KB-PF had increased prediction accuracy. The proposed method is applied to real-world data and outperforms the KB-VM in terms of prediction accuracy. The KB-PF and the KB-VM were both tested on 60-hour predictions, yielding average distance error rates of 52 km and 73 km, respectively (based on the fused data).

Zissis *et al.* (2015) proposed a cloud-based architecture, capable of perceiving and predicting the behaviour of multiple vessels, i.e. their spatial location, SOG, and COG, by implementing an ANN. The proposed ANN was designed as a cloud-based application, with the ability to overlay predicted short and long term vessel trajectories, including the behaviours on an interactive map. The time horizon for the short-term predictions is 15 minutes, and for the long term predictions, 75 and 150 minutes, respectively. The ANN learns vessel patterns in specific geospatial regions, utilising any historic AIS data recorded in a given region. The performance of the ANN was evaluated using real AIS data. The proposed method had good prediction accuracy in terms of the vessel location prediction and SOG prediction. However, it had difficulty predicting the COG since the vessels which were considered, rapidly changed course. When comparing the prediction accuracies for the short and the long term predictions of the ANN, the long term accuracies were worse in comparison to the short term prediction accuracies. In both cases, the ANN was able to recognise and predict the overall behaviour of the vessels, which include the prediction of the location, SOG and COG. This observation is expected, as any changes in a vessel's COG or incorrect prediction thereof will lead to an increase in prediction error as more time passes.

Zissis *et al.* (2016) created a method to accurately predict future coordinates of a vessel by using ANNs. Different types of model pre-processing and construction techniques are implemented, and the ANN was trained using Historic AIS data. The model learns (adapts) in real-time as the data changes whilst predicting with a prediction time horizon of up to 15 minutes. The worst error rate recorded was 4.75%, measured as the percentage difference between the actual and predicted output.

Hexeberg *et al.* (2017) presented a method that uses historic AIS data to predict future locations of vessels. The method is called Single Point Neighbour Search (SPNS). The method does a close neighbour (CN) search by extracting historic observations within a certain radius of the current vessel's spatial location. Vessels in the CN set that do not adhere to prespecified COG range values are removed. Using the CN set, the median COG and SOG value is calculated. Using the median COG and SOG, the predicted longitude and latitude is calculated. The method predicts in constant distance intervals, where the SOG is used to calculate the time passed between two observations. The method can confidently predict with a time horizon of up to 15 minutes.

Jaskolski (2017) implemented a Discrete Kalman Filter (DKF), to predict future locations of vessels. The DKF, in the context of vessel coordinate prediction, constantly adjusts itself for an improved prediction as new observations are observed. It is assumed that a vessel fitted with an AIS sensor will not constantly send updates. The DKF consists of two sets of equations: predictor and measurement update equations. Burger *et al.* (2020) showed that there is no significant improvement in the prediction accuracy by using a DKF over a Linear Regression Model (LRM) if both are used to predict linear trajectories.

Dalsnes *et al.* (2018) proposed a prediction method that utilises Gaussian Mixture Modelss (GMMs). The use of GMMs allows for the measurement of the degree of uncertainty and handles multimodality. The proposed method is built on a method called the Neigbor Course Distribution Method (NCDM) developed by Hexeberg (2017). The NCDM was developed to improve on some of the shortfalls of the SPNS method created by Hexeberg *et al.* (2017). The SPNS prediction output can be seen as a list of states forming a single trajectory, whereas the output from NCDM is a tree of states which forms multiple trajectories. Each trajectory calculated by the NCDM is calculated similarly to that of the SPNS method. The NCDM has a higher complexity than that of the SPNS but allows for the prediction of trajectories in several branched sea lanes with an uncertainty measure attached. The predicted future position calculated by the NCDM is given by a number of $J^{max}$ points taken from the desired level of the prediction tree. The proposed method extends the NCDM by fitting a GMM to the predicted points resulting in a probabilistic model of the future position. The Expectation Maximisation (EM) algorithm is used to fit the GMM, which will fit the maximum likelihood GMM for the given points (Dalsnes *et al.*, 2018). The GMM, as in the case of the NCDM, allows for the prediction of trajectories branching over several sea lanes. Thus, the predicted future position's distribution can be seen as a multimodal distribution. The prediction horizon of the proposed method is 5-15 minutes. Dalsnes *et al.* (2018) recorded a median Root Mean Square Error (RMSE) of 290 m and 675 m for the 5 and 15 minute prediction horizon, respectively.

Kim and Lee (2018) proposed a deep NN model called the Ship Traffic Extraction Network (STENet) to predict medium-term ($20 - 30$ minutes) and long-term ($40 - 50$ minutes) traffic in a so-called caution area. The caution area is an area identified by VTS operators in which numerous vessel route intersections exist with high traffic flow. The proposed method was designed

to minimise the risk in such areas, reducing the probability of vessel collisions or groundings. A vessel's SOG and COG can suddenly change, resulting in many parameters that have to be tracked by the VTS operators. STENet was trained on historic AIS data. STENet was organised into a hierarchical architecture. The hierarchical architecture consists of a front-part (feature extraction) and rear-part (prediction). The front-part consists of two feature extraction modules: vessel movement vectors and vessel attributes, i.e. caution area, vessel length, vessel destination, pilot embarkation, and vessel type. The first module consists of a CNN, and the second consists of five Fully-Connected Convolutional Neural Networks (FCNNs) each receiving an associated attribute. The extracted features are concatenated and fed into the prediction module's rear-part. The purpose of feature extraction at the front-part of the architecture is to prevent cross-talking between unrelated attributes. The rear-part consists of a FCNN, which predicts the number of vessels that will be in a caution area. Four prediction models were compared, namely Dead Reckoning, the Support Vector Regression (SVR), the STENet and the VGGNet (a model proposed by Simonyan and Zisserman (2015)). Kim and Lee (2018) showed when comparing the STENet to the SVR, that the STENet was 50.65% more accurate when compared on short term predictions, and 57.65% more for long-term predictions. The error was calculated using the Mean Absolute Percentage Error (MAPE), which calculates the difference between the true and predicted trajectory as a percentage.

Virjonen *et al.* (2018) proposed a trajectory prediction method using $k$-Nearest Neighbours ($k$-NNs). The idea behind the method is to compare the target vessel[11] with other vessels that historically resided in the same geospatial area. The predicted trajectory is estimated with a $k$-NN model by finding $k$ matching routes from the historical AIS data with similar behavioural characteristics as the target vessel. The measure of similarity allows one to identify vessels that were close to each other historically with a similar SOG in the given geospatial area. For the experimental design, Virjonen *et al.* (2018) made use of two predefined spatial areas, where the distance from the starting point of the spatial areas to the endpoint was 60 km and 120 km, respectively. Nested leave-one-out cross-validation was used to evaluate the performance of the method. Acceptable prediction accuracy was achieved for the use case (prediction of an emissions control vessel's trajectory, which is mostly linear). The prediction accuracy was between $3-5$ minutes for the 60 km geospatial area and $7-12$ minutes for the 120 km geospatial area.

[11] In this thesis, a *target vessel* refers to the vessel of interest. It is the vessel whose state is being estimated, or whose trajectory is being predicted (the vessel a specific algorithm is applied to).

Rong *et al.* (2019) proposed a probabilistic trajectory prediction model based on a Gaussian Process (GP). The GP can describe the uncertainty of a vessel's future position along the predicted route, using continuous probability density functions (pdfs). The GP is a non-parametric data-driven Bayesian model. The proposed model decomposes a vessel's movement into longitudinal and latitudinal directions, and a positional pdf is fitted for both and independent of each other. Rong *et al.* (2019) also proposed a route-fitted coordinate system that allows them to better describe a vessel's motion. The route-fitted system consists of a centerline of a specific traffic route. The traffic route denotes an area with specific traffic flow historically, similar to a highway where there is a clear region where vessels have travelled. The traffic route was obtained from historic vessel trajectories of AIS data[12]. The centerline is calculated by the Dynamic Time Warping (DTW) algorithm (Müller, 2007). The route-fitted system converts the LAT and LON coordinates to their route-fitted space. An acceleration pdf is created based on the analysis of historic AIS data. The longitudinal motion pdf, of a vessel, is calculated as an integral of its acceleration (Rong *et al.*, 2019). A GP regression (GPR) model is then applied to the acceleration pdfs, to estimate the pdf of a vessel's position in the latitudinal direction. The hyperparameters of the GP are obtained from historic vessel trajectories and is regarded as prior knowledge of a vessels position derived from historic traffic flow. Clear traffic flow can be seen in the historic data used by Rong *et al.* (2019), allowing them to estimate the pdfs. The GPR model can forecast the pdf over the future trajectory of a vessel, resulting in a pdf that describes the future position of a vessel and the associated certainty thereof. The aforementioned pdf is two-dimensional and is created by combining the two one-dimensional pdfs of the respective movement directions (Rong *et al.*, 2019). The proposed model can be applied in real-time and can predict a pdf of a vessel's future location and the uncertainty thereof. The real-time prediction is made by iteratively updating the prior pdfs as new observations are observed from the vessel. Given a prediction horizon of 120 minutes, the latitudinal and longitudinal errors were 800 m and 1700 m, respectively. A prediction horizon of 60 minutes is recommended when using the proposed method, as Rong *et al.* (2019) reported that a longer prediction horizon results in an exponential increase in prediction error.

Forti *et al.* (2020) made use of a Deep Learning (DL) Neural Network (NN) approach to predict trajectories of vessels. A sequence-to-sequence Recurrent Neural Network (RNN) model, that

[12]The vessels Rong *et al.* (2019) used for their proposed model, all travelled within the traffic route and are in the same spatial range of the historic data. They also only considered one route to showcase the proposed method.

utilises a Long Short-Term Memory (LSTM) encoded-decoder Recurrent Neural Network (RNN) is proposed. Historic AIS data is used to train the LSTM model. The method aims to learn the predictive distribution of maritime traffic patterns using historic AIS data. Learning the predictive patterns enables the model to predict more accurately. It was shown that the model could predict more accurately than the OU process, given a time window of 20 observations.

Liu *et al.* (2020) proposed an Online Multiple outputs Least-Squares Support Vector Regression model based on a Selection Mechanism (SM-OMLSSVR). The SM-OMLSSVR is based on an offline Multiple outputs Least-Squares Support Vector Regression (MLSSVR) model, introduced in Xu *et al.* (2013). The Least-Squares Support Vector Regression (LSSVR) model, is a Support Vector Regression (SVR) model (proposed by Vapnik (1999)), where the inequality constraints are replaced with equality constraints by the addition of slack variables (proposed by Suykens *et al.* (2002), also see Saunders *et al.* (1998), Suykens and Vandewalle (1999)). For in depth information on SVR see Hastie *et al.* (2009). The SM-OMLSSVR is a modified version of the MLSSVR to an online hybrid model that can incorporate new updates received from a vessel when deemed necessary. The incorporation of new updates depends on the selection mechanism. The implemented selection mechanism can be explained as follows: once a new observation is observed the error of the MLSSVR model is calculated, and if the error is not within allowable range (as set by Liu *et al.* (2020)) the LSSVR model is used to add the new samples to the training set of the MLSSVR, in turn updating the model. If the error is in an acceptable range, the offline model (MLSSVR) continues to predict. The selection model essentially decides when to incorporate new information, turning the MLSSVR into an online method until an acceptable error is obtained. The SM-OMLSSVR was compared to an RNN LSTM model, a NN and a traditional LSSVR model in terms of prediction accuracies. The methods were tested on six sample trajectories whose trajectories' time ranges between three to six hours. Prediction errors within $[5-30]$ m were recorded. One should note that the SM-OMLSSVR's training set updates as new observations are received for improved prediction accuracies.

Murray and Perera (2020) presented a novel dual linear autoencoder (AE) approach to predict a vessel's trajectory. The method predicts a future trajectory using historic AIS data. The proposed method predicts an entire trajectory, where all the vessel states are predicted simultaneously. The method estimates a latent distribution of the possible future trajectories of the target vessel. By sampling from the latent distribution, multiple trajectories can be predicted together with their

uncertainties. For a prediction horizon of 30 min, the median prediction error was found to be 2.5%, where the error is calculated as the distance from the mean $\mu_j$ of the corresponding distribution $\boldsymbol{P}_j$ and the true position $(p_j)$ at that state. $\boldsymbol{P}_j$ is a normal distribution fitted to all the predicted positions of each predicted state. The error is presented as a percentage of the actual distance travelled by the target vessel.

Suo *et al.* (2020) proposed a DL framework that uses a Gate Recurrent Unit (GRU) model to predict vessel trajectories. A series of trajectories together with vessel information (spatial location, SOG and COG) is extracted from AIS data. The main trajectories are then derived by applying the Discovering Clusters in Large Spatial Databases with Noise (DBSCAN) algorithm introduced by Ester *et al.* (1996). A trajectory information correction algorithm is also applied, which utilises symmetric segmented-path distance. The applied algorithm eliminates large amounts of redundant data and optimises incoming trajectories. A GRU model is then applied to predict real-time vessel trajectories. Historic AIS data was used to train and test the model. The GRU was compared to an LSTM model. The GRU had improved computational time efficiency with similar prediction accuracy to that of the LSTM. Suo *et al.* (2020) noted that the current model's recursive nature is computationally expensive when applied to long-distance trajectory prediction.

Wang *et al.* (2020) proposed a vessel berthing trajectory prediction model based on a Bidirectional Gated Recurrent Unit (Bi-GRU). The proposed model learns from historic AIS data of vessels located in the Tianjin port in China. Wang *et al.* (2020) extracts the LAT, LON SOG, COG and time from the historical data to build a set of vessel berthing trajectories to train, validate and test the proposed model. The Bi-GRU model requires the previous four consecutive trajectory points of the target vessel. The Bi-GRU shows promising results when compared to an LSTM and GRU for short distance predictions in port waters and has a smaller error and higher accuracy.

Xiao *et al.* (2020) did an extensive review of maritime knowledge mining and traffic forecasting technologies. An LRM is compared to several non-linear approaches. Three broad categories of non-linear algorithms are considered: machine learning approaches, knowledge-based approaches and control theory assisted methods. The predictions range from long to short-term predictions. It is also shown that more complex methods are more accurate at a higher computational cost.

Xu (2020) presented a context-based trajectory prediction algorithm utilising LSTM networks.

Real-valued target trajectories are converted into discrete path sets from historical data, and distinctive patterns are clustered hierarchically. Two models are compared, a RNN consisting of one LSTM and another RNN consisting of $k$ LSTMs. In the RNN with $k$ LSTMs, an LSTM is created for each distinct path that exists. The proposed LSTM network outperformed the standard LSTM network.

Zhang *et al.* (2020) proposed an AIS data-driven model for vessel final destination prediction and not the trajectory leading up to the final destination. The proposed model is based on a Random Forest (RF), which utilises the similarity between a vessel's current trajectory and its historic trajectory to predict the vessel's final destination. The historic destination, whose associated trajectory is the most similar to that of the current vessel's trajectory, will be returned by the RF, as the predicted final destination. To validate the performance of the proposed model, 11 trajectory similarity based measurement methods and two different decision strategies were compared and implemented. Eight conventional similarity measures were used, namely Fréchet Distance (Har-Peled *et al.*, 2002), Edit distance with real penalty (Chen and Ng, 2004), Edit distance on a real sequence (Chen *et al.*, 2005), Longest Common Subsequence (Gruber *et al.*, 2009), Dynamic Time warping (Wang *et al.*, 2013), Hausdorff Distance (Magdy *et al.*, 2015), Discrete Fréchet (Gudmundsson and Valladares, 2014), and Symmetrized segment-path distance (Besse *et al.*, 2016). Three ML similarity measures were used, namely the Naive Bayes classifier, Multi-layer perceptron, and the Independently RNN. The two decision strategies were the: maximum similarity-based decision strategy and the port frequency-based decision strategy. The proposed method achieved a prediction accuracy of 70% for port-based prediction and 80% for city-based predictions when implemented on a five-day prediction horizon.

Alizadeh *et al.* (2021) proposed three novel prediction methods based on historic AIS data . The first method proposed is a Point-based Similarity Search Prediction (PSSP), which was inspired by Wijaya and Nakamura (2013). The historical points are measured in terms of their spatial location, SOG and COG. The second method proposed is called Trajectory-based Similarity Search Prediction (TSSP), where each recorded AIS trip is regarded as a trajectory. The PSSP is a point-based method, whereas the TSSP is a trajectory-based method. Lastly, a Trajectory-based Similarity Search Prediction is proposed using an RNN LSTM (TSSPL). Alizadeh *et al.* (2021) point out that vessel movement is affected by external movements such as wind, waves, and sea

currents. The PSSP and the TSSP are not able to account for these external factors. Another RNN LSTM model was, therefore, built to take this into account (i.e. TSSPL). The TSSPL has an additional input, a measure of similarity between trajectories (similar to what was done by Tang *et al.* (2019)). Prediction horizons of 10, 20, 30, and 40 min were considered on 89 different vessel trajectories. The TSSPL had decreased errors of 53.6%, 54.2%, 55.8%, and 55.2% at the different time intervals when compared to the PSSP and TSSP. When comparing the prediction errors over time the TSSP recorded a reduced error of 40.85% when compared to the PSSP. The TSSPL had a reduced error of 23% when compared to the TSSP.

Bautista-Sánchez *et al.* (2021), presented a sample method to select historic AIS data on vessel-specific routes, named Select Best AIS Data in Prediction Vessel Movements and Route Estimation (PreMovEst). The method's goal is to optimise the computational performance of vessel spatial location prediction and for real-time trajectory estimation. The method makes use of an ANN and AIS data from a VTS database. The PreMovEst method consists out of four components divided into two different stages: training and discovering. The first stage consists of three components: acquiring AIS data, the processing thereof, and selecting the best routes through a statistical behavioural selection process, called Chi-squared selection. The final stage consists of finding and predicting the position of vessel movements, consisting of two techniques. The first is an RNN with LSTM, where historical data is used as continual input streams for trajectory prediction. The second technique employs a Multivariate Imputation by Chained Equations (MICE), a statistical method used for handling missing data (Bautista-Sánchez *et al.*, 2021). MICE allows for the approximation of the full predicted path. The PreMovEst method had reasonable prediction accuracies of between $80.5 - 84\%$.

Gao *et al.* (2021) proposed a novel Multistep Prediction Long Short-Term Memory (MP-LSTM) model which was inspired by the Trajectory Proposal Network for Motion Prediction (TPNet) framework created by Fang *et al.* (2020*a*). TPNet is a multi-step prediction method proposed for vehicle trajectory prediction and assumes that future trajectories are cubic splines. TPNet utilises a Convolutional Neural Network-based Encoder and Decoder to predict future curves (trajectories). TPNet predicts two points, the curvature (pivot) point and an endpoint. Gao *et al.* (2021) combined the TPNet and an LSTM; using the LSTM to predict the curvature control point. The determination of the aforementioned point is regarded as a regression problem rather than a classi-

fication problem in TPNet, as such, it was relabelled to "support point". The trajectory endpoint is obtained under the hypothesis that the values generated from the reference trajectory and the current trajectory are approximately equal. The reference trajectory is created from the vessel's motion parameters using historical data. Gao *et al.* (2021) also proposed an automatic generation method for the reference trajectory. The proposed prediction method predicts in constant time displacement intervals and was created to overcome the two disadvantages that trajectory prediction algorithms have; namely, they make use of complex relationship mappings and require large amounts of data to work. The method was verified on four different navigational states: the straight, turning, acceleration, and deceleration motion states. The accuracy was measured using the final and average displacement error. The method was compared to four other methods; namely, the dual linear AE proposed by Murray and Perera (2020), the EKF, the Support Vector Regression Model, and an LSTM. Overall, the MP-LSTM outperformed the various other models in terms of accuracy for each of the four navigational states.

Murray and Perera (2021) presented a DL framework to aid in regional ship behaviour prediction, with the help of historic AIS data. Regional refers to the region in which the framework will be deployed, and the historic patterns and behaviours observed in the region. The framework was created to aid in collision avoidance for improved maritime transport safety. The presented framework consists of grouping historical vessel behaviour in given geospatial areas into clusters, where each cluster contains vessel trajectories with similar behavioural characteristics. A model is created for each unique cluster, modelling the unique behaviour specific to each cluster. The clustering is performed by a variational Recurrent AE (Murray and Perera, 2020), and a Hierarchical Density-Based Spatial Clustering of Applications with Noise algorithm (Campello *et al.*, 2013). The historical behaviour of a vessel is classified as belonging to the most likely cluster behaviour based on the softmax distribution (also known as the normalised exponential distribution). The softmax gives a probability distribution over all the possible classes and is used in this study to identify a distribution over all the vessel behavioural clusters the trajectory segments most likely belong to. A probability is assigned to each cluster. Murray and Perera (2021) implemented a threshold when considering a cluster as a behavioural cluster, only clusters with a softmax output over 0.1 (a probability of 10%) were considered. More information on the softmax function and its characteristics can be found in Géron (2019). Each local model consists of a sequence-to-sequence

RNN (Forti *et al.*, 2020). The presented DL framework requires the historic trajectory of a vessel as an input; it then predicts the most likely future trajectory. It utilises the aforementioned most likely cluster whilst predicting. The presented framework was compared to a global model given in Forti *et al.* (2020). A global model refers to a model trained on all the available data in a given region. In the presented framework, a local model was trained on the data belonging to each cluster identified, using the same sequence-to-sequence architecture as used by Forti *et al.* (2020). Given a prediction horizon of 30 min, a root mean squared error of 436 m and 576 m were observed for the presented and the global models, respectively.

Wang and He (2021) proposed a Generative Adversarial Network (GAN) with Attention Module and Interaction Module (GAN-AI) to predict the trajectories of multiple vessels. The idea behind the GAN-AI is to improve the ability of the network to extract effective data at multiple time points. The GAN-AI infers all future vessel trajectories simultaneously when they are in the same spatial area and is trained by competition for better convergence. An interactive module was designed to extract group motion features[13] of multiple vessels (Wang and He, 2021). The interaction model was designed in such a way to extract group monition features of multiple vessels, to achieve better performance in complex vessel behavioural patterns. The proposed method was tested with prediction horizons of 90s and 180s, and compared to an LSTM, a plain GAN and a Kalman model. The prediction accuracy, respectively, improved by 20%, 24% and 72% when the GAN-AI is compared to the aforementioned models. The trajectory prediction accuracy was evaluated by utilising the average distance and final distance errors.

The purpose of this thesis is to develop a simple first-order model that incorporates historic AIS data in such a way that does not require a lot of model training and parameter tuning. The developed model should be pragmatically easy to implement from an operational point of view, computationally efficient and accurate compared to other models with similar programmatic complexity. The overall goal of the thesis is to contribute a simple model to the literature as the majority of recent developments are rather complex models (as outlined in the literature review).

Figure 2.3, depicts a visual representation of the citations received for each study in the literature review. The circles' sizes are logarithmically proportional to the number of citations received. The colours of the circles can be interpreted as follows: circles in black $>$ 100 citations, blue $>$ 50

[13]Motion features are any features that describe the movement of a vessel, i.e. SOG, COG, vessel status and ROT.

citations, green $> 15$, orange $> 5$, and yellow $\geq 0$ citations.

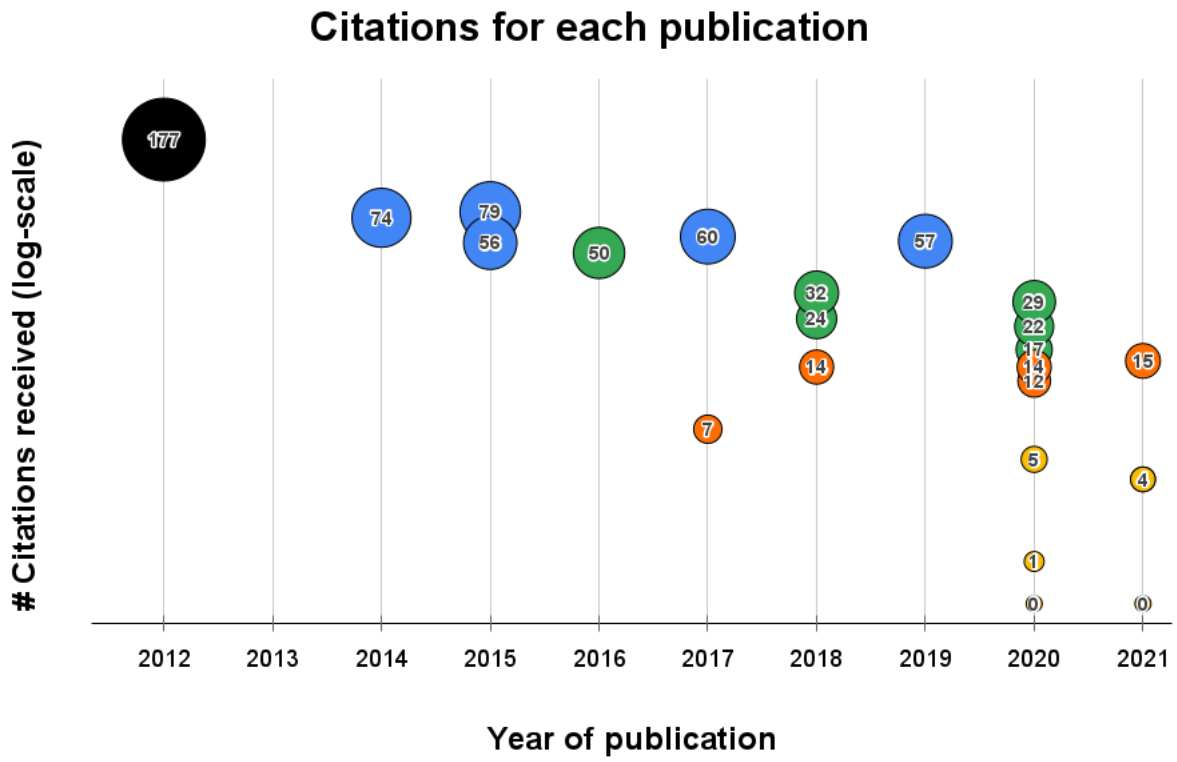


Figure 2.3: Visual representation of the citations each study mentioned in the literature review achieved[14]. The circle size is proportionate to the number of citations. Note that the exact citations matching the study can be seen in Table 2.6, which also provides a summary.

[14]The symbol #, in the Figure 2.3, denote the count.

| Study | Prediction Type (Trajectory / State / End destination) | ML Methods implemented | Prediction Horizon upperlimit | *a priori* incorporation | Number of Citations |
|---|---|---|---|---|---|
| Perera *et al.* (2012) | Trajectory, State | EKF, ANN | 50 s | Training | 104 |
| Pallotta *et al.* (2014) | Trajectory | OU | Hours (Problem depended) | Training, Prediction | 74 |
| Mazzarella *et al.* (2015*a*) | Trajectory | PF, KB-PF, KB-VM | 60 hours | Training, Prediction | 75 |
| Zissis *et al.* (2015) | Trajectory, State | ANN | 150 min | Training | 56 |
| Zissis *et al.* (2016) | Trajectory | ANN | 15 min | Training | 50 |
| Hexeberg *et al.* (2017) | Trajectory | *N.A.* | 15 min | Prediction | 60 |
| Jaskolski (2017) | Trajectory | DKF | 380 s | Initialisation | 7 |
| Dalsnes *et al.* (2018) | Trajectory | GMM | 15 min | Training, Prediction | 24 |
| Kim and Lee (2018) | State | CNN, FCNN, SVR | 50 min | Training | 32 |
| Virjonen *et al.* (2018) | Trajectory | k-NN | 120 km | Prediction | 14 |
| Rong *et al.* (2019) | Trajectory | GP | 60 min | Training, Prediction | 57 |
| Forti *et al.* (2020) | Trajectory | DL-NN, LSTM, OU | 20 observations | Training | 17 |
| Liu *et al.* (2020) | Trajectory | SVR, LSTM, NN | 6 hours | Training, Prediction | 1 |
| Murray and Perera (2020) | Trajectory | AE | 30 min | Training, Prediction | 22 |
| Suo *et al.* (2020) | Trajectory | DL, GRU, LSTM | *N.A.* | Training | 12 |
| Xiao *et al.* (2020) (Survey) | Trajectory, State | LRM, OU, ANN, EKF, SVM, RF, etc.[15] | 20 hours | Training, Prediction | 29 |
| Xu (2020) | Trajectory | LSTM | *N.A.* | Training | 0 |
| Zhang *et al.* (2020) | End destination | RF, Naive Bayes, NN, RNN | 5 days | Training | 14 |
| Alizadeh *et al.* (2021) | Trajectory | LSTM | 40 min | Training | 10 |
| Bautista-Sánchez *et al.* (2021) | Trajectory | ANN, LSTM | *N.A.* | Training, Prediction | 0 |
| Gao *et al.* (2021) | Trajectory and State | LSTM, AE, SVR, EKF | *N.A.* | Training, Prediction | 4 |
| Murray and Perera (2021) | State and Trajectory | RNN, DL | 30 min | Training | 3 |
| Wang and He (2021) | Trajectory | GAN | 90 s | Training | 1 |

Table 2.6: A summary of the literature review and the number of each study received. The counts are based on the number of citations recorded on 15 November 2021 (provided by Google Scholar).

[15]See Xiao *et al.* (2020), as they did an extensive review on multiple ML algorithms

Note that although many of the studies we included in our literature review make use of a variety of Machine Learning (ML) techniques, we do not discuss these techniques any further in this thesis. An in-depth presentation of all these methods is beyond the scope of the thesis. We refer the reader to the following textbooks if any clarification on any of the techniques mentioned in this literature review is sought: "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" (Géron, 2019), "Deep Learning" (Goodfellow *et al.*, 2016), "Artificial Intelligence: A Modern Approach" (Russell and Norvig, 2002), and "The Elements of Statistical Learning: Data Mining, Inference, and Prediction" (Hastie *et al.*, 2009). Having said this, it should, however, also be noted that we do discuss the particular ML/AI techniques that are relevant to our study in more detail in Chapter 4.

## 2.3 SUMMARY

The goal of this chapter is to provide background information on the different types of transponder based tracking technologies that currently exist. The chapter should enable the reader to have a more in-depth understanding of the different technologies, how they work, and what they are specifically used for. Complementary technologies to transponder based technologies are also presented. We compared the technologies, showcasing their drawbacks and advantages. A summary of the technology comparison is given in Table 2.5. From this comparison, it was concluded that AIS is the primary system being used for vessel tracking. Therefore we did a deep dive into AIS, more specifically methods used to predict vessel locations and trajectories utilising AIS data currently in the literature. The literature review should provide the reader an in-depth overview of vessel prediction methods, how they work, the type of methods they utilise, their prediction accuracies and time horizons. In the next chapter, the dataset used for our experiments is introduced together with SMs.

# CHAPTER 3

# DATASET

In this chapter, we introduce the AIS dataset we use for all our experiments and the pre-processing steps applied. We will also introduce the SMs, what they are and how they were constructed from the AIS data.

Ray *et al.* (2019) published an opensource dataset that consists of AIS messages from vessels that traversed the Celtic Sea, the North Atlantic Ocean, the English Channel and the Bay of Biscay (France) from October 2015 to March 2016. The publicly available dataset has lead to various studies being published, a few of these studies are on the following topics:

- Anomaly detection: (Anneken *et al.*, 2018), (Machado, 2018), and (Iphar *et al.*, 2020).
- Classification: (Li *et al.*, 2020) and (Pitsikalis *et al.*, 2021).
- Trajectory compression: (Fikioris *et al.*, 2020*a*) and (Fikioris *et al.*, 2020*b*).
- Trajectory clustering: (Theodoropoulos *et al.*, 2019) and (Tampakis, 2020).
- Trajectory prediction method comparison: (Burger *et al.*, 2020).

In Figure 3.1, a visual representation of the data's geospatial range can be seen. The specific vessels we tested the presented methods in this paper on, contained within the dataset, are summarised in Appendix C in Table C.1 and C.2.

| **Type** | **Details** |
|---|---|
| **Number of Observations** | 19 035 630 |
| **Number of attributes** | 9 |
| **File Type** | Comma Separated Values File (.csv) |
| **File Size** | 1.04 GB |

Table 3.1: Dataset characteristics and information

In Table 3.1, important information pertaining to the dataset is given, and in Table 3.2, the dataset attributes used is introduced.

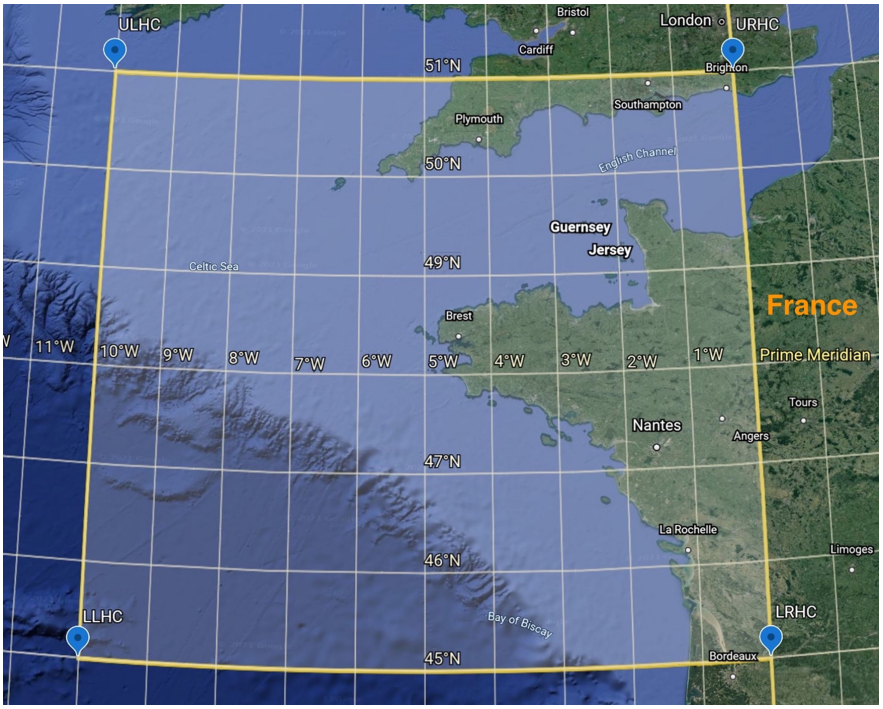


Figure 3.1: Visual expand of the spatial range of the data in Ray *et al.* (2019) is given - Courtesy of Google Earth 2021. Let the abbreviations and their coordinates be defined as: Upper left-hand corner (ULHC) (10, 51), Upper right-hand corner (URHC) (0, 51), Lower right-hand corner (LRHC) (0, 45), and Lower left-hand corner (LLHC) (10, 45).

| **Attributes Description** | | |
|---|---|---|
| **Description** | **Measurement Unit** | **Attribute Range** |
| **MMSI** | | 9-digit values |
| **Latitude** | DD.dddd | [-10.00, 0.00] |
| **Longitude** | DD.dddd (UTM) | [45.00, 51.00] |
| **COG** | Degrees | 0° - 360° |
| **SOG** | knots (kt) | 0° - 110° |
| **Date Time Stamp** | Coordinated Universal Time (UCT) | [ 2015-10-01 00:00:00, 2016-03-31 23:59:59 ] |
| **Ship Type** | | [10, 99] |

Table 3.2: Dataset Attributes

## 3.1 DATASET CLEANING

Data cleaning steps are applied to the dataset to make it more usable for the experiments implemented throughout this thesis. In this section, we discuss some of the steps that we take to perform the cleaning (pre-processing).

The first step is to remove observations from the dataset that do not adhere to the following criteria:

- Ship type within the inclusive interval $[70, 89]$, where Cargo vessel types are in $[70, 79]$ and Tankers in $[80, 89]$
- SOG $> 0.5$ kt, removing stationary observations, including stationary vessels experiencing drift due to currents and other natural phenomena due to being anchored.
- SOG $< 60$ kt, observations with high speeds are likely outliers as Cargo and Tanker vessels move at relatively low speeds. If more than 60 kt is observed for Cargo or Tanker vessels, it is most likely due to technical errors in the recorded AIS data.

The remainder of the data is grouped according to the vessel MMSIs and sorted in ascending order according to the recorded time of each observation. All trajectories with less than 20 observations or those that span less than five minutes in total are removed. The data spans over a period of six months, as denoted in Table 3.2, which implies that there will be more than one trajectory for a given vessel at different time periods and spatial locations. The data cleaning is done to remove any non-sensical data that may skew models applied to it.

### 3.1.1 Dataset Statistics

In this subsection, we discuss some dataset statistics, giving the reader an overview of the aforementioned data cleaning steps' effect on the dataset.

#### *3.1.1.1 Vessel Types*

In Table 3.3, a breakdown of the number of unique vessels belonging to each vessel type is shown.

| Vessel Type | Count | % Observations |
|---|---|---|
| **Cargo** | 2871 | 57.29 |
| **Tankers** | 1119 | 22.33 |
| **Fishing** | 30 | 0.60 |
| **Passanger** | 64 | 1.28 |
| **Other** | 927 | 18.5 |
| **Total** | **5011** | **100** |

Table 3.3: A breakdown of the unique vessels per type.

As part of this thesis, we only focus on Cargo and Tanker vessels, utilising 79.62% of the dataset as-is. A further breakdown of Cargo and Tanker vessel types is listed in Table 3.4.

The effect of cleaning the data is presented in Table 3.4. We observe that 44% of all observations

| Vessel Type | Description | # Observations | | | | # Unique MMSIs | |
|---|---|---|---|---|---|---|---|
| | | Before Cleaning | <0.5kt | >= 60kt | After Cleaning | Before Cleaning | After Cleaning |
| **70** | Cargo | 1 673 130 | 726 534 | 5 320 | 941 276 | 1550 | 1529 |
| **71** | Cargo Hazard A | 228 341 | 27 571 | 37 | 200 733 | 236 | 236 |
| **72** | Cargo Hazard B | 10 067 | 0 | 0 | 10 067 | 19 | 19 |
| **73** | Cargo Hazard C | 15 770 | 6 666 | 1 | 9 103 | 10 | 10 |
| **74** | Cargo Hazard D | 27 201 | 1 918 | 0 | 25 283 | 32 | 31 |
| **75** | Cargo | 30 | 0 | 0 | 30 | 2 | 2 |
| **76** | Cargo | 40 | 0 | 0 | 40 | 4 | 4 |
| **77** | Cargo | 1 695 | 75 | 0 | 1620 | 4 | 4 |
| **78** | Cargo | 17 | 0 | 0 | 17 | 1 | 1 |
| **79** | Cargo | 220 793 | 59 059 | 64 | 161 670 | 306 | 301 |
| **80** | Tanker | 518 253 | 319 012 | 341 | 198 900 | 369 | 365 |
| **81** | Tanker Hazard A | 64 795 | 25 816 | 2 | 38 977 | 73 | 72 |
| **82** | Tanker Hazard B | 317 878 | 249 446 | 2 | 68 430 | 97 | 86 |
| **83** | Tanker Hazard C | 12 046 | 1 313 | 1 | 10 732 | 32 | 32 |
| **84** | Tanker Hazard D | 47 634 | 13 448 | 3 | 34 183 | 38 | 37 |
| **85** | Tanker | 0 | 0 | 0 | 0 | 0 | 0 |
| **86** | Tanker | 0 | 0 | 0 | 0 | 0 | 0 |
| **87** | Tanker | 0 | 0 | 0 | 0 | 0 | 0 |
| **88** | Tanker | 2 698 | 71 | 2 | 2 625 | 1 | 1 |
| **89** | Tanker | 228 812 | 68 182 | 2 | 160 628 | 200 | 200 |
| **Total** | | **3 369 200** | **1 499 111** | **5 775** | **1 864 314** | **2 974** | **2930** |
| **% of Original Observations** | | **100%** | **44%** | **0.17%** | **55%** | **100%** | **98.5%** |

Table 3.4: Observation breakdown before and after cleaning.

recorded were considered as stationary observations. When a vessel is anchored or moored, they still send AIS messages albeit at a reduced frequency. There were 0.17% observations with an SOG of more than 60 kt which are outliers for the use case of Cargo and Tanker vessels. After all the cleaning is done, enforcing the first two cleaning steps of Section 3.1 (SOG and ship type) only 55% of the original dataset remains, containing 2974 unique vessel MMSIs. After dropping vessels with less than 20 observations or a trajectory that spans less than 5 min, the remainder is 2930 unique vessel MMSIs; a 1.48% reduction. The final dataset contains 45% of the original data points.

#### *3.1.1.2 SOG breakdown*

In Figure 3.2 and 3.3, the number of observations belonging to an SOG interval is shown as a percentage. Figure 3.2 is associated with the original dataset, while Figure 3.3 is associated with the cleaned dataset. Both figures denotes the % observations per SOG interval.

When looking at Figure 3.2, we see that the majority of observations are within the $0-5$ kt interval.

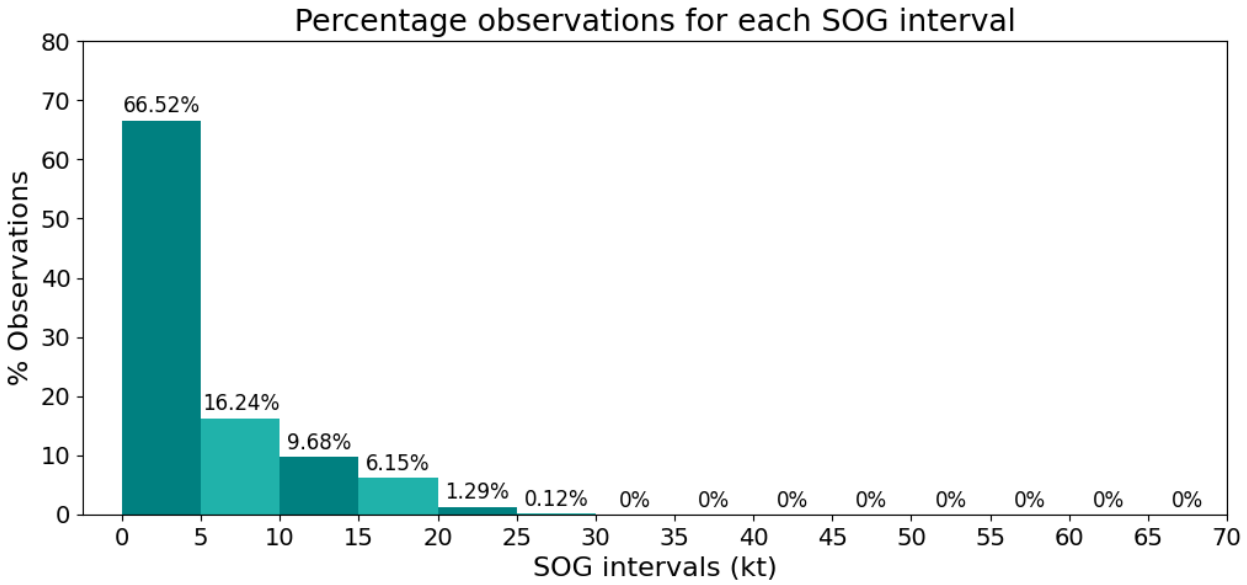


Figure 3.2: Distribution of SOG **before** dataset pre-processing.

This is due to numerous stationary observations being present in the dataset. In Figure 3.3, we see the distribution changes dramatically where the $5-10$ kt interval now containing the majority observations. In Figure 3.2, 99% of the observations fall within the $0-30$ kt interval and $< 1\%$ in the $30-70$ kt interval (0% in the Figures represent a percentage $> 0\%$). A similar observation follows from Figure 3.3, 99% of the observations fall between $0-30$ kt and $< 1\%$ of the observations in the $30-60$ kt interval. The distribution of observations as seen in Figure 3.3 at different speed intervals, are more representative of a dataset with Cargo and Tanker vessels that are in movement.

#### *3.1.1.3 Longitude and Latitude*

In Figure 3.4 below, we see a spatial distribution map (SDM) of all the observations in the dataset (including all vessel types). An SDM can be thought of as a matrix, where each cell in the matrix denotes the count of vessels recorded in a specific geospatial area over a period of time. Figure 3.5 denotes an SDM of only Cargo and Tanker vessels of the cleaned dataset. The spatial dimensions of the SDM are noted in Table 3.2 (with $1250 \times 1250$ cells). Each cell of the SDM denotes a square spatial area of longitude and latitude size $0.008 \times 0.008$ degrees.

The SM projections throughout this section were all generated with the Open Source Geographic Information System software (QGIS), a free and opensource tool (QGIS Development Team, 2021).

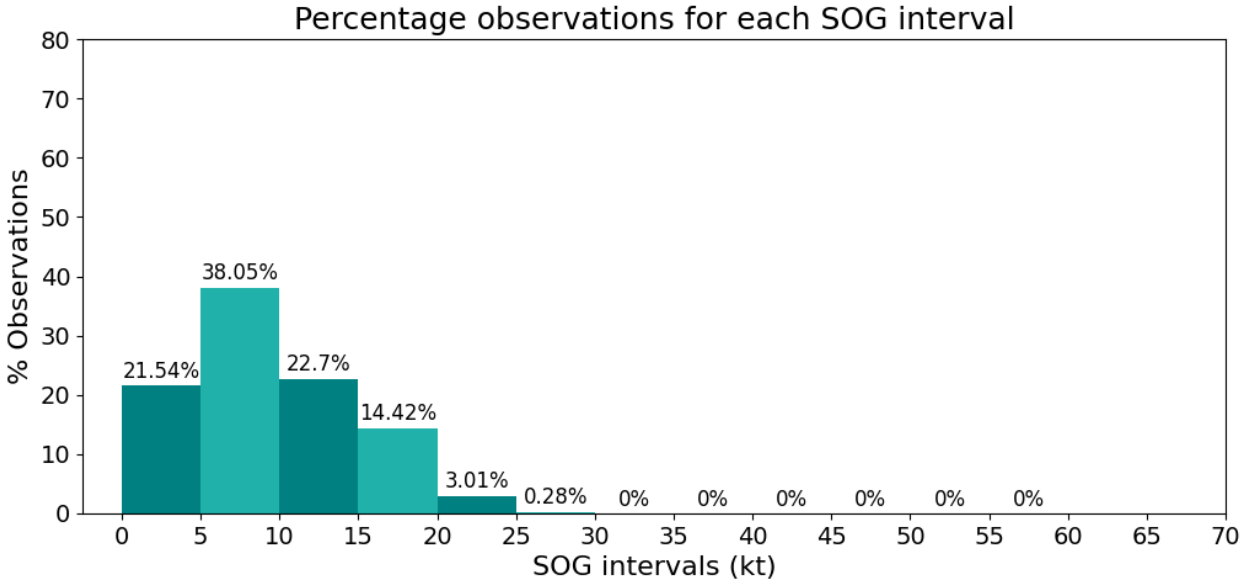


Figure 3.3: Distribution of SOG **after** pre-processing.

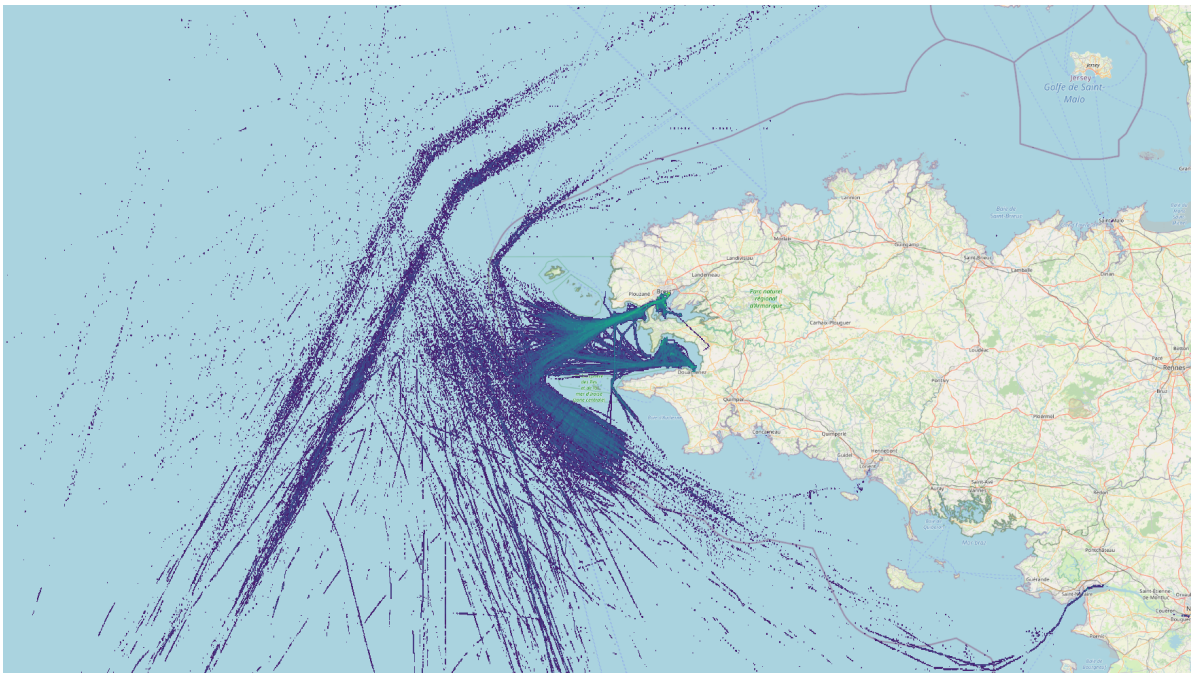


Figure 3.4: An SDM projection of the dataset **before** any cleaning steps have been applied.

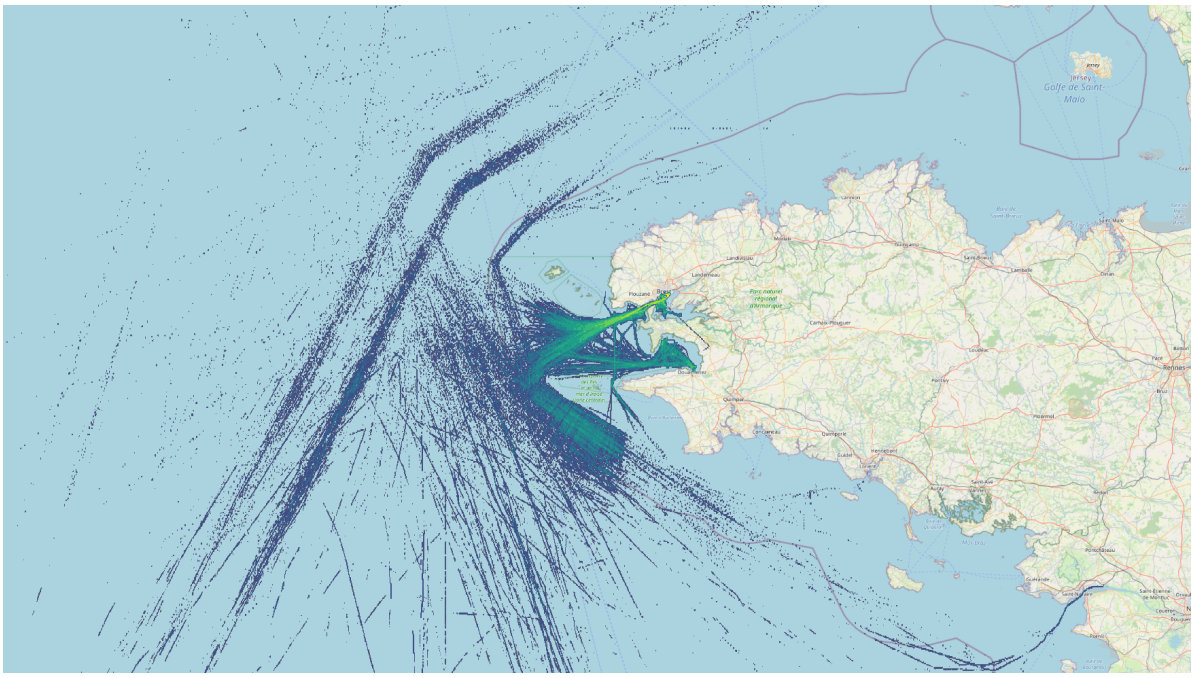


Figure 3.5: An SDM projection of the dataset **after** the cleaning steps have been applied.

Figures 3.4 and 3.5, represent the projection of the SDMs on a map. The yellowish areas denote an area containing many observations and are most likely ports or harbours or are close to them. Various vessels are in close proximity for extended periods of time in these regions. The count of observations per cell is dependent on a vessel's SOG. A slower moving vessel will record more observations over the same spatial area than a faster moving vessel, as AIS messages are reported at regular time intervals.

Figure 3.4 represents an SDM before any cleaning steps have been applied to the dataset (the dataset as is), where Figure 3.5 denotes the SDM after the cleaning steps have been applied. With respect to Figure 3.5 we can see there is a clear difference, with some paths being more visible/prominent (represented by yellow) compared to Figure 3.4.

Figure 3.6 and 3.7 depicts a zoomed in view of Figure 3.4 and 3.5, respectively. The zoomed in view is of a harbour in Brest, France. Comparing Figure 3.7 to 3.6, a clear path highlighted in yellowish tint is now visible. The visibility of the path is due to the removal of stationary observations. We observe that there are more observations closer to landmass. This is due to better T-AIS coverage, and due to vessels moving slower to avoid collisions with the shore or other vessels. In Table 3.5 an example[1] of the time it takes for vessels to stop is shown, the explaining why vessels move slower closer to shorelines.

| **Vessel Type** | **Speed (kt)** | **Average Stopping Distance** |
|---|---|---|
| **Cargo** | 10 | 2 NM |
| **Tanker** | 10 | 4.5 NM |

Table 3.5: Example of Cargo and Tanker Stopping Distance

In the next section we introduce a trajectory interpolation algorithm, to have a more representative set of vessel trajectories and their true paths. The interpolation allows us to generate SMs that will be more representative of the actual paths vessels took, and increase the representation of vessels in areas of weak signal coverage between vessels and T-AIS stations.

[1]The example was obtained from: `https://www.knowledgeofsea.com/stopping-distance-turning-circle-ships-manoeuvring/`

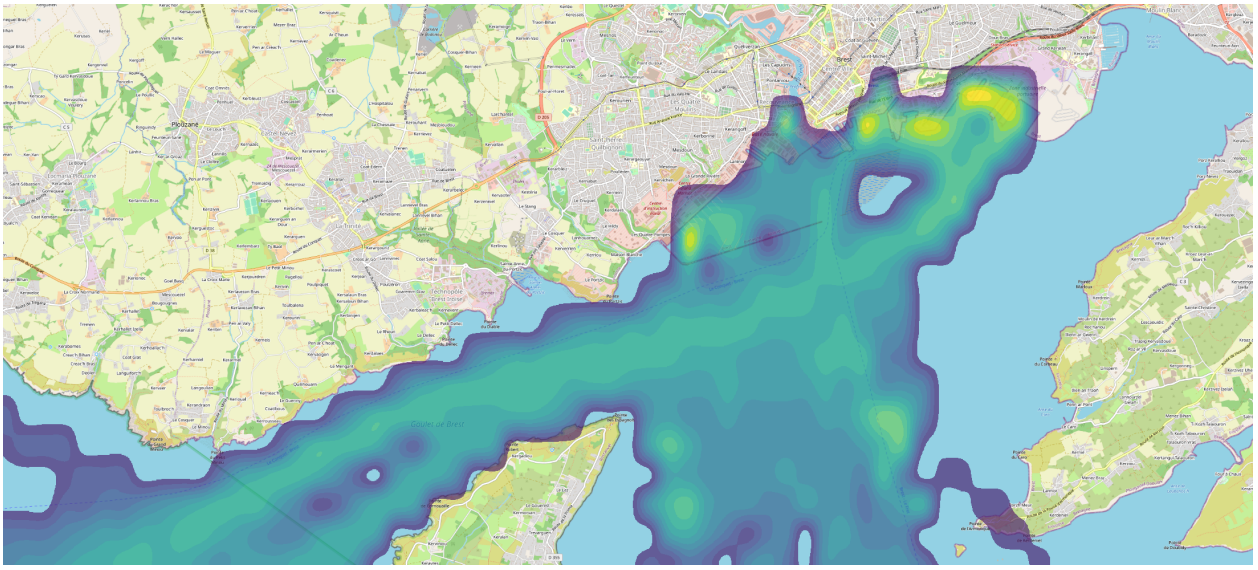

Figure 3.6: A zoomed in version of Figure 3.4 of the harbour in Brest, France.

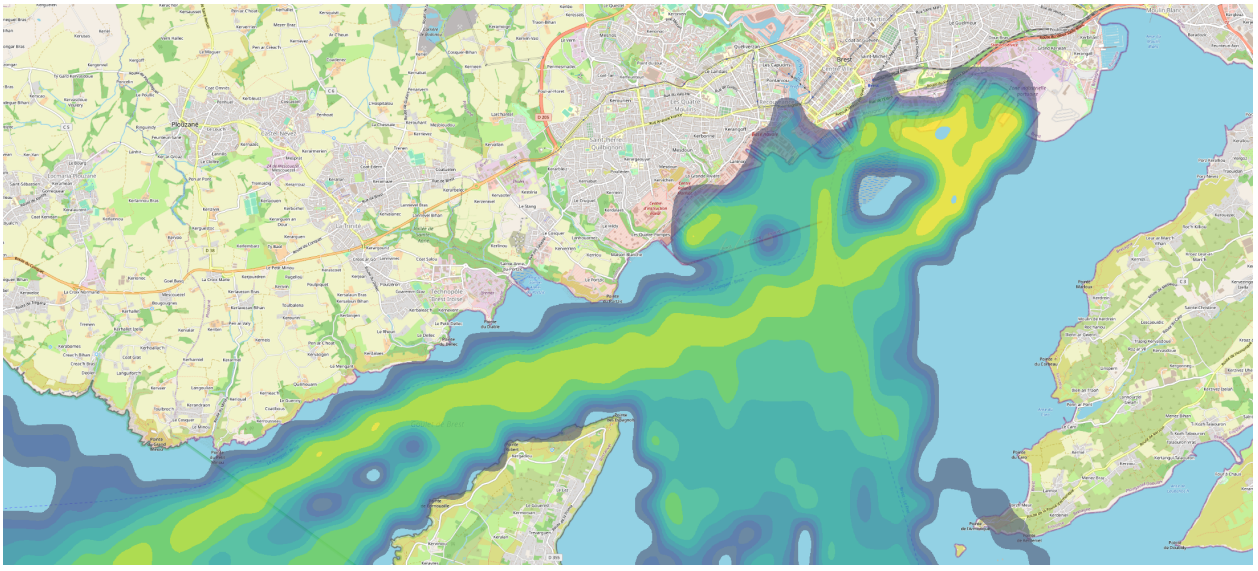

Figure 3.7: A zoomed in version of Figure 3.5 of the harbour in Brest, France.

## 3.2 SPATIAL MAPS

We define a Spatial Map (SM) as a two-dimensional grid that spans the earth's surface. Each cell in the two-dimensional grid is associated with a specific range of latitudinal ($\phi$) and longitudinal ($\lambda$) coordinates.

In this thesis our SMs are set up from the observations recorded in the dataset introduced earlier, which spans a latitude and longitude of $\phi \in [45°, 55°]$ and $\lambda \in [-10°, 0°]$, respectively. We only use the observations from the Cargo and Tanker vessels as these are our target vessels. SMs help us to visualise the historical paths where vessels traversed and identify highways - paths commonly travelled historically (Grobler and Kleynhans, 2019). The SMs can be thought of as a data reduction method, where each cell in the SM represents a summary of a specific spatial area. SMs in this thesis will be used to represent *a priori* information and can be used as an input to an algorithm.

For the sake of simplicity we assume throughout this thesis that the SMs are square maps, meaning the SM's width ($n_{l_c}$) and length ($n_{l_r}$) have the same number of cells. Let the dimensions of the SMs be defined by $n_{l_c} \times n_{l_r}$, where $[n_{l_c}, n_{l_r}] \in \mathbb{N}$ [†]. The dimensions of the SMs, are based on the dataset introduced in Section 3. The SMs used throughout the thesis have the dimensions of $1250 \times 1250$ cells which translates to $10° \times 10°$ square degrees, based off of the range of the LAT and LON. The spatial resolution for each cell, therefore, is equal to $0.008° \times 0.008°$ square degrees. An extract of the upper left-hand corner of an artificially created SM is depicted in Figure 3.8. The figure denotes the number of recorded observations in a cell, within a longitude and latitude range based on the historic AIS data. On the $x$- and $y$-axis of Figure 3.8 the longitudes and latitudes are denoted for the created SM.

[†]$\mathbb{N}$ refers to natural numbers, where $\mathbb{N} = \{1, 2, 3, 4, ...\}$.

| Latitude \ Longitude | 0.008 | 0.016 | 0.024 | ... |
|---|---|---|---|---|
| 0.000 | 0 | 0 | 2 | 0 |
| 0.008 | 0 | 2 | 5 | 0 |
| 0.016 | 1 | 2 | 5 | 0 |
| 0.024 | 2 | 5 | 1 | 0 |
| ⋮ | | | | |

Figure 3.8: SDM matrix example extract.

Let the Haversine distance between two coordinates be defined as:

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right), \tag{3.1}$$

where

- $r$, represents the mean radius of the earth ($\approx 6371$km).
- $\phi_1$ and $\phi_2$ represent the latitudinal coordinates of two observations, point one and two respectively.
- $\lambda_1$ and $\lambda_2$ represent the longitudinal coordinates of two observations, point one and two respectively.

Given Equation 3.1, the area associated with each cell is roughly $0.89 \times 0.89$ square kilometres, where 0.89 km $\approx$ 0.48 NM. When we first constructed the SMs, our first observation was that the SMs were very sparse when created from only Cargo and Tanker vessels after pre-processing. Therefore we interpolated vessel trajectories allowing for more representative SMs. We present three SMs in the remaining next few sections. The first SM that we discuss is the Vessel Counts

SM ($\boldsymbol{K}$), followed by the Course Over Ground SM (COGSM, represented by $\boldsymbol{\Psi}$) and the COG standard deviation (COG SD) SM ($\boldsymbol{\Sigma}$).

### 3.2.1 Vessel Interpolation for Spatial Maps

After pre-processing the data, an augmented dataset is constructed by interpolating between observations for each unique vessel MMSI, creating less sparse trajectory observations for each vessel. This augmented dataset is then used to construct physically realistic and usable SMs of the dataset in question. The original dataset is too sparse (as is) for a meaningful and usable SM to be created from it. The sparsity of the original data is due to several reasons. Observational interpolation was only performed in the following cases:

- The time difference between the two observations is no longer than six hours.
- The distance between the two observations is within 15 km and
- The distance between observations is no smaller than the size of one grid cell. The grid cell size used was $0.88\text{km} \times 0.88\text{km}$. This specific constraint prevents the over-representation of a grid cell, only adding one observation to a cell if the interpolated trajectory passes through it.

We make use of a linear interpolation model; we used the software package scikit-learn (Pedregosa *et al.*, 2011). The LAT, LON and SOG were interpolated if the cases above were met. Gaussian filtering is also used to obtain smoother versions of the SMs.

The COG of the interpolated and recorded observations is calculated via:

$$\psi_t = \arctan\left(\frac{\phi_{t-1} - \phi_t}{\lambda_{t-1} - \lambda_t}\right) \tag{3.2}$$

We do not make use of the recorded COG values present in the dataset, as these values are inaccurate. The COG was calculated based on the LAT and LON. Having a new augmented dataset with a better representation of the historical locations and trajectories, we can create representative SMs.

### 3.2.2 Vessel Counts SM ($\boldsymbol{K}$)

The vessel counts SM, represents the number of observations recorded within each grid cell. Each grid cell in $\boldsymbol{K}$ records the number of observations that were historically within the longitude and latitude range of that specific cell. The cell counts will be higher in spatial areas where vessels are sailing slower than those where vessels are sailing faster. Figure 3.9 shows the logarithmically scaled SDM of the vessel counts for each cell.

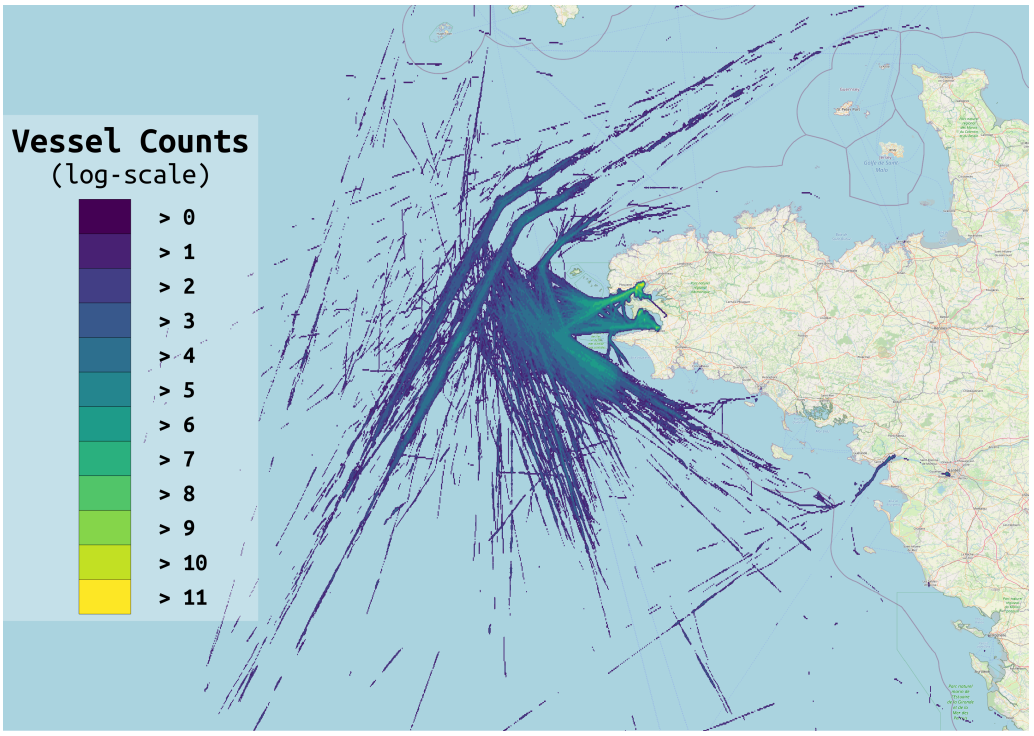


Figure 3.9: Vessel Counts SM $\boldsymbol{K}$. Each colour denotes the number of observations per cell on a continuous log-scale.

### 3.2.3 COG ($\boldsymbol{\Psi}$) and COG SD ($\boldsymbol{\Sigma}$) SMs

The second SM that we introduce is the COG SM, represented by $\boldsymbol{\Psi}$. Each cell of $\boldsymbol{\Psi}$ represents the mean COG value recorded in that cell. COG is measured in degrees, periodic in $[0° - 360°]$ and will always be positive ($\psi_{(i_\phi, i_\lambda)_j} > 0$).

The third and last SM that we introduce is the COG standard deviation (COG SD) SM, represented

by $\mathbf{\Sigma}$. The COGSD SM supplements the COG SM. The value of each cell is calculated as follows:

$$\mathbf{\Sigma}_{i_\phi, i_\lambda} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(\psi_{(i_\phi, i_\lambda)_j} - \mathbf{\Psi}_{i_\phi, i_\lambda}\right)^2} \tag{3.3}$$

where,

- $n$, represents the number of observations observed in a cell as determined by $K$.
- $i_\phi$, $i_\lambda$, represents the index values associated with $\phi$ and $\lambda$ respectively on the SM grid.
- $\mathbf{\Psi}_{i_\phi, i_\lambda}$, represents the mean COG at a specific index value on the SM grid.
- $\psi_{(i_\phi, i_\lambda)_j}$, represents the $j^{\text{th}}$ COG value in the cell with index values $i_\phi, i_\lambda$.
- $\mathbf{\Sigma}_{i_\phi, i_\lambda}$, refers to the COG standard deviation at index value $i_\phi, i_\lambda$.

Loosely speaking, the entries in $\mathbf{\Sigma}$ can be interpreted as how "confident" we ought to be in the corresponding entry in $\mathbf{\Psi}$. Higher cell values in $\mathbf{\Sigma}$ mean that historically many vessels were travelling in different directions, as the SD is higher. $\mathbf{\Sigma}$ allows us to have a type of uncertainty value associated with the *a priori* information.

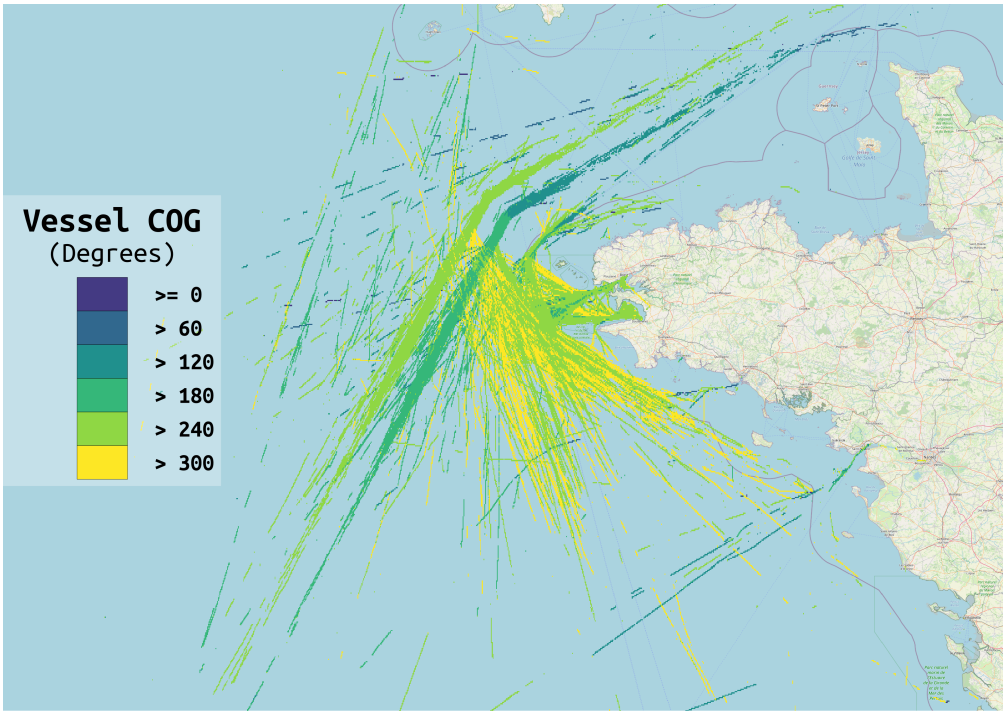


Figure 3.10: Course over Ground SM $\mathbf{\Psi}$, the colours represent observations from $0 - 360°$ as indicated by the legend.

In Figure 3.10, we can see a visual representation of $\mathbf{\Psi}$. Looking at the figure, we can see that Cargo and Tanker vessels move in a specific direction, in certain geographic locations. Two distinct highways are clearly visible close to the centre of Figure 3.10 (Grobler and Kleynhans, 2019). A highway is a route that many vessels traverse. The light green highway is used by Cargo and Tankers to travel upwards (North), while the blue/aqua-green highway is used to travel downward (South).

In Figure 3.11, we see a visual representation of $\mathbf{\Sigma}$, whose analytic expression is given in Equation 3.3. The standard deviation (SD) of the areas that contain more traffic in different directions are larger than those containing less traffic (this is especially true for areas surrounding harbours). Yellowish colours represent higher SD values. The highways mentioned, however, are not associated with high SD values. This implies that these highways are highly directional, as shown in Figure 3.11.

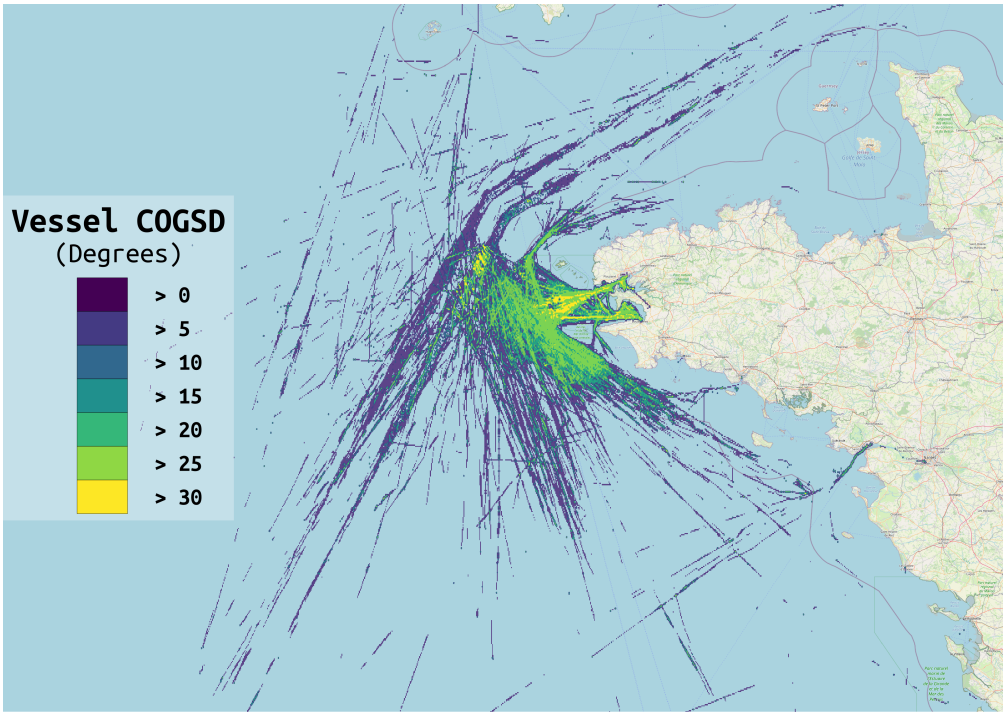


Figure 3.11: A visual projection of the COG Standard Deviation SM $\mathbf{\Sigma}$. Each colour represents a standard deviation range, as noted by the legend.

## 3.3 TRAJECTORY VISUALISATION

Throughout this thesis we refer to linear and non-linear trajectories. In Figure 3.12, we provide a visualisation of these two types. The vessels visualised are from Table C.1 and C.2 in Appendix C. Furthermore, to put the size of the trajectories in perspective, the non-linear trajectory spans 82.16 nautical miles (NM). The linear trajectory span 20.05 NM.

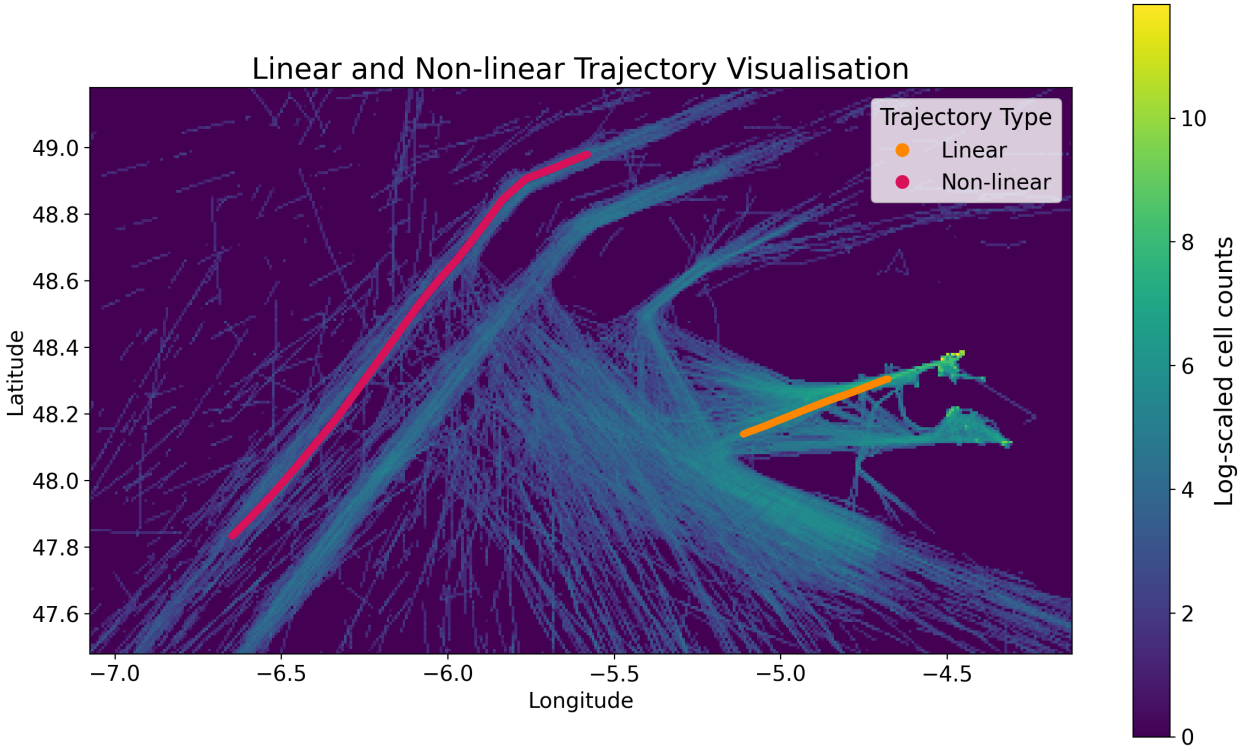


Figure 3.12: Visualisation of a Linear and Non-linear trajectory, belonging to vessels with MMSIs 304927000 and 304805000 respectively.

To showcase the size of the vessels we are working with, in Figure 3.13 we show an example of a Cargo vessel's (Maersk Edinburgh class) length compared to the Eiffel tower's height.

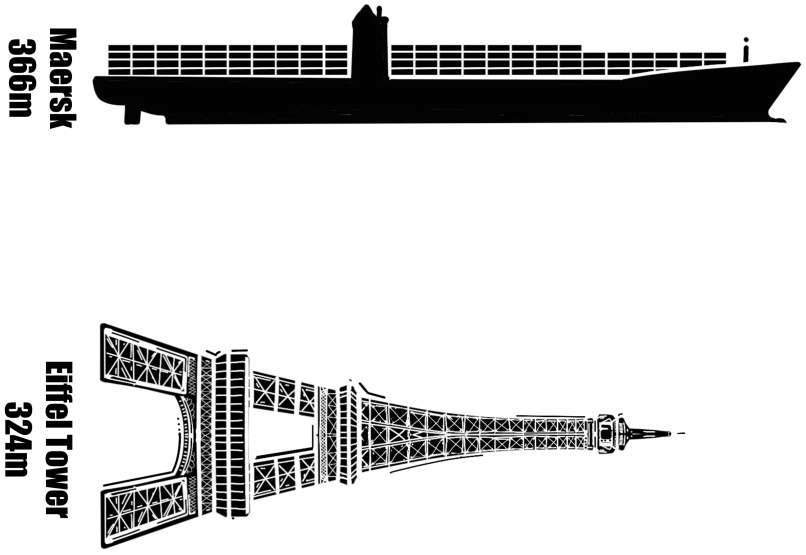


Figure 3.13: A visualisation of a Cargo vessel's length compared to the Eiffel Tower's height.

## 3.4 SUMMARY

In this chapter we introduced the dataset used throughout this thesis to test and evaluate all the implemented algorithms. The dataset contains AIS data, from the region depicted in Figure 3.1. The dataset cleaning (pre-processing) steps are also discussed. We introduce SMs, as they are generated from the aforementioned AIS dataset. With respect to the SMs, trajectory interpolation steps are also presented to allow for more representative SMs. We also showcase the difference between a linear and non-linear trajectory. The next chapter introduces the trajectory prediction methods implemented in this thesis, the non *a priori* and *a priori* methods that utilise the dataset and SMs introduced in this chapter.

# CHAPTER 4

# METHODOLOGY

In this chapter we discuss the methods that we implemented to conduct our experimental analysis. Each method will be discussed in detail. First we discuss the unit conversions necessary for the methods to work. Then we present two AIS prediction algorithms that do not make use of *a priori* information, namely the Discrete Kalman Filter (DKF) in Section 4.2, and the Linear Regression Model (LRM) in Section 4.3. We end the chapter with two AIS prediction algorithms that uses *a priori* information whilst predicting, namely the Single Point Neighbour Search (SPNS) in Section 4.4, and a novel method the Linear Regression Model with *a priori* COG information (LRMAC) in Section 4.5.

## 4.1 UNIT CONVERSIONS

In order for the algorithms to run as intended, the unit conversion of some of the attributes in the cleaned AIS dataset has to take place. In particular, the recorded coordinates and the SOG observations.

### 4.1.1 Coordinates

The non *a priori* methods, the DKF and the LRM, require the conversion of the LAT and LON coordinates from degrees (DD.dddd) to Universal Transverse Mercator* (UTM) coordinates (measured in metres).

The *a priori* methods, the LRMAC and the SPNS, do not convert the coordinates recorded by the AIS transmitter, and they are used as-is, i.e. in degrees (DD.dddd).

### 4.1.2 Speed

Let $V_t'$ represent the SOG recorded by an AIS transmitter. Let the unit conversion of the SOG from knots to metre per second $(m/s)$ be shown by:

*UTM refers to the projection of coordinates on a two-dimensional plane. LAT and LON denote coordinates on a sphere in our case, the earth.

$$V_t = 0.514 \cdot V_t' \tag{4.1}$$

The DKF, LRM, and SPNS make use of the SOG in this converted form. The LRMAC, however, requires a different conversion which is discussed in more detail in Section 4.5.1.

## 4.2 DISCRETE KALMAN FILTER (DKF)

Rudolph E. Kalman invented the Kalman Filter (KF) in 1960 and published the following seminal paper: "A new approach to Linear Filtering and Prediction Problems" (Kalman, 1960). Numerous applications of the KF exist. In this section, a KF, more specifically the DKF, is used to predict the trajectory of a vessel in regular time intervals. In Appendix A, the DKF is discussed in detail.

The DKF works by continuously estimating the state of a system and the uncertainty of the estimation made. All calculations are done on a recursive basis by evaluating two sets of equations: predictor- and measurement update equations. The two sets of equations allow for online prediction.

The predictor equations estimates (predicts) the current state of a system and the uncertainty thereof. The measurement update equations update the estimate by using an observation. The error between the estimated observation and the true observation (when observed), is calculated and incorporated in such a way to determine the accuracy of the DKF (through the covariance matrices).

The aforementioned alternating procedure of the DKF is depicted mathematically in Figure 4.1, which was originally created by Welch *et al.* (1995). The meanings of all the symbols are discussed in the following section.

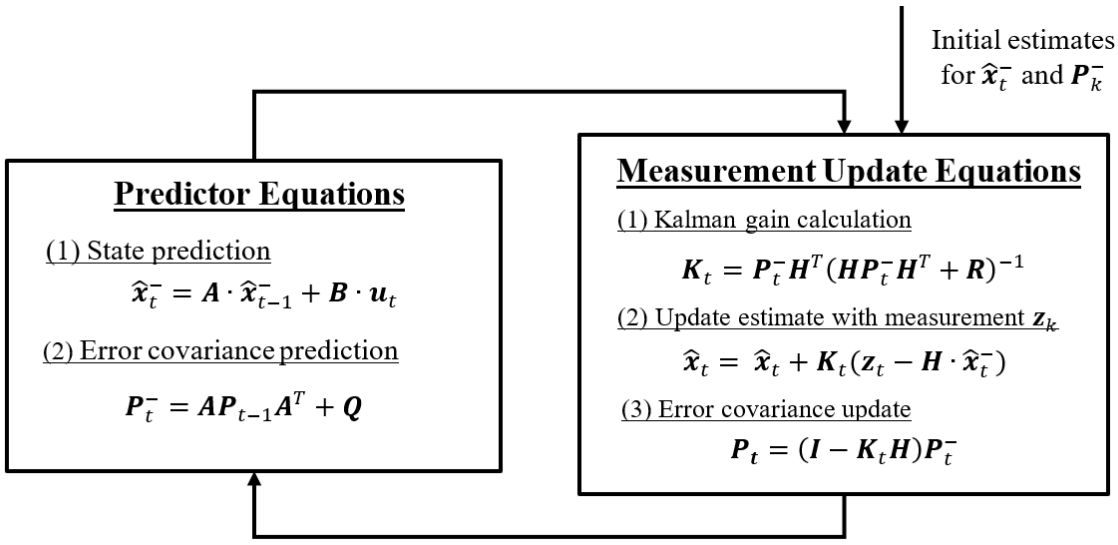


Figure 4.1: The DKF operation visualised.

The DKF assumes that both the process ($\boldsymbol{Q}$) and the observation ($\boldsymbol{R}$) noise are normally distributed. It is important to note that the DKF requires regular sampled data points across time (Kalman (1960)).

### 4.2.1 DKF for vessel trajectory prediction

In this section, we describe the DKF approach used in Jaskolski (2017) to predict vessel trajectories. Let $\boldsymbol{x}_t$ denote the state vector of the DKF. For the AIS use case, it is defined as follows:

$$\boldsymbol{x}_t = \begin{bmatrix} x_t & y_t & V_t \cdot \cos(\psi_t) & V_t \cdot \sin(\psi_t) \end{bmatrix}^T, \tag{4.2}$$

where,

- $x_t$ and $y_t$ are the true LON and the LAT of a vessel at time-step $t$.
- $\psi_t$ and $V_t$ are the true COG (measured in degrees) and SOG (kt) of a vessel at time step $t$, respectively.
- The superscript $(\cdot)^T$ refers to taking the transpose.

The predictor equations of the DKF are:

$$\hat{\boldsymbol{x}}_t^- = \boldsymbol{A} \cdot \hat{\boldsymbol{x}}_{t-1} + \boldsymbol{B} \cdot \boldsymbol{u}_t, \tag{4.3}$$

and

$$\boldsymbol{P}_t^- = \boldsymbol{A} \cdot \boldsymbol{P}_{t-1} \cdot \boldsymbol{A}^T + \boldsymbol{Q}. \tag{4.4}$$

The variables in the above equations are defined below:

- $\hat{\boldsymbol{x}}_t^-$ denotes the predicted state estimate using all observations up until time-step $t-1$.
- $\hat{\boldsymbol{x}}_{t-1}$ denotes the updated state estimate using all observations up until time-step $t-1$.
- $\boldsymbol{A}$ is the state transition matrix.
- $\boldsymbol{B}$ is the output matrix, a constant matrix that forms part of the predicted state.
- $\boldsymbol{u}_t$ is the control variable vector at time-step $t$.
- $\boldsymbol{P}_t^-$ denotes the predicted error covariance estimate using all observations up until time-step $t-1$.
- $\boldsymbol{P}_{t-1}$ denotes the updated error covariance estimate using all observations up until time-step $t-1$.
- $\boldsymbol{Q}$ denotes the covariance of the process noise.

The measurement update equations of the DKF are:

$$\boldsymbol{K}_t = \boldsymbol{P}_t^- \cdot \boldsymbol{H}^T \cdot (\boldsymbol{H} \cdot \boldsymbol{P}_t^- \cdot \boldsymbol{H}^T + \boldsymbol{R})^{-1}, \tag{4.5}$$

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_t^- + \boldsymbol{K}_t \cdot (\boldsymbol{z}_t - \boldsymbol{H} \cdot \hat{\boldsymbol{x}}_t^-), \tag{4.6}$$

and

$$\boldsymbol{P}_t = (\boldsymbol{I} - \boldsymbol{K}_t \cdot \boldsymbol{H}) \cdot \boldsymbol{P}_t^-. \tag{4.7}$$

The variables in the above equations are defined below:

- $\boldsymbol{H}$ denotes the transformation matrix, allowing for the mapping of our state-space to our observed space.

- $\boldsymbol{R}$ denotes the covariance associated with the observational noise.
- $\boldsymbol{K}_t$ denotes the Kalman gain at time-step $t$.
- $\hat{\boldsymbol{x}}_t$ denotes the updated state estimate using all observations up until time-step $t$.
- $\boldsymbol{z}_t$ denotes the observed measurement at time-step $t$.
- $\boldsymbol{P}_t$ denotes the updated error covariance estimate using all observations up until time-step $t$.
- The difference $(\boldsymbol{z}_t - \boldsymbol{H} \cdot \hat{\boldsymbol{x}}_k^-)$ is called the residual, the difference between the actual measurement and the predicted measurement.

If there are no measurements at a specific time-step the measurement update equations are skipped, and the predictor equations are re-evaluated. When using the DKF for AIS trajectory prediction, Jaskolski (2017) suggested to initialise our state vector and error covariance as follows:

$$\hat{\boldsymbol{x}}_0 = \begin{bmatrix} 0\,m & 0\,m & 0\,m/s & 0\,m/s \end{bmatrix}^T, \tag{4.8}$$

and

$$\boldsymbol{P}_0 = \begin{bmatrix} \sigma_x^2 & \sigma_y^2 & \sigma_x^{V\,2} & \sigma_y^{V\,2} \end{bmatrix}^T \cdot \begin{bmatrix} \sigma_x^2 & \sigma_y^2 & \sigma_x^{V\,2} & \sigma_y^{V\,2} \end{bmatrix}. \tag{4.9}$$

In the above equation:

$$\sigma_x = 10,\ \sigma_y = 10,\ \sigma_x^V = 0.3,\ \sigma_y^V = 0.3. \tag{4.10}$$

Jaskolski (2017) also proposed that we use the following matrices and vector to evaluate Equation 4.3 – 4.7, they are chosen specifically for the problem at hand:

- For $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{pmatrix} 1 & 0 & \Delta k_t & 0 \\ 0 & 1 & 0 & \Delta k_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.11}$$

  In the above equation, $\Delta k_t$ denotes the time difference between each predicted time step (regular time steps). Therefore, $\boldsymbol{A}$ is responsible for obtaining the displacement of a vessel after $\Delta k_t$ seconds have passed. In this thesis we assumed $\Delta k_t = 1$s (predicting each second).

- For $\boldsymbol{B} \cdot \boldsymbol{u}_k$:

$$\boldsymbol{B} \cdot \boldsymbol{u}_t = \begin{bmatrix} a_x \cdot \frac{\Delta k_t^2}{2} & a_y \cdot \frac{\Delta k^2}{2} & a_x \cdot \Delta k_t & a_y \cdot \Delta k_t \end{bmatrix}^T, \tag{4.12}$$

where

$$a_x = \frac{V_k \cdot \cos(\psi_t + \varphi_t \cdot \Delta k_t) - V_{k-1} \cdot \cos(\psi_{k-1})}{\Delta k_t}, \tag{4.13}$$

$$a_y = \frac{V_k \cdot \sin(\psi_t + \varphi_t \cdot \Delta k_t) - V_{k-1} \cdot \sin(\psi_{k-1})}{\Delta k_t}, \tag{4.14}$$

$$\boldsymbol{B} = \begin{bmatrix} \frac{\Delta k_t^2}{2} & 0 & 0 & 0 \\ 0 & \frac{\Delta k_t^2}{2} & 0 & 0 \\ 0 & 0 & \Delta k_t & 0 \\ 0 & 0 & 0 & \Delta k_t \end{bmatrix}, \tag{4.15}$$

and

$$\boldsymbol{u}_k = \begin{bmatrix} a_x & a_y & a_x & a_y \end{bmatrix}^T. \tag{4.16}$$

In the above equations:

  - $a_x$ and $a_y$ denote the acceleration of a vessel in the $x$ and $y$ direction respectively measured in m/s$^2$.
  - $\boldsymbol{B}$ can be seen as a constant valued matrix, and the product thereof with $\boldsymbol{u}_t$ converts the accelerations to displacement and speed in the $x$ (LON) and $y$ (LAT) direction.
  - $\varphi_t$ denotes the ROT (degrees/$s$) of a vessel at time-step $t$.

- For $\boldsymbol{H}$:

$$\boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.17}$$

- For $\boldsymbol{R}$:

$$\boldsymbol{R} = \begin{bmatrix} \sigma_x^2 & 0 & 0 & 0 \\ 0 & \sigma_y^2 & 0 & 0 \\ 0 & 0 & \sigma_x^{V^2} & 0 \\ 0 & 0 & 0 & \sigma_y^{V^2} \end{bmatrix} \tag{4.18}$$

- For $\boldsymbol{Q}$:

$$\boldsymbol{Q} \cong \boldsymbol{P}_0. \tag{4.19}$$

The matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ of the DKF are used in the predictor and measurement update parts of the DKF respectively. For the DKF to perform well, care should be taken in choosing the initial values of the aforementioned matrices.

#### *4.2.1.1 The Kalman gain*

Let the Kalman gain be denoted by $\boldsymbol{K}_t$ (see Equation 4.5). $\boldsymbol{K}_t$ is responsible for weighting the measurement error (see Equation 4.6), also known as the residual. The gain is influenced by matrices $\boldsymbol{R}$, $\boldsymbol{P}_t^-$ and indirectly by $\boldsymbol{Q}$ (see Equation 4.4). The influence of the aforementioned matrices is further explained in this section.

The measurement covariance matrix $\boldsymbol{R}$, is a constant for the vessel trajectory prediction use case we are considering in this thesis. Smaller values of $\boldsymbol{R}$ will result in the actual measurement ($\boldsymbol{z}_k$), as well as the residual, being weighted more ("trusted" more). Smaller values will also weigh the predicted measurement $\boldsymbol{H} \cdot \hat{\boldsymbol{x}}_k^-$ less ("trusted" less). The same will be true for the reverse case.

The *a priori* estimate error covariance matrix $\boldsymbol{P}_t^-$ is calculated by Equation 4.4. Smaller values of $\boldsymbol{P}_t^-$ will result in $\boldsymbol{K}_k$ assigning less weight in the actual measurement $\boldsymbol{z}_t$ (i.e. less "trust" is given to the measurement), while the predicted measurement $\boldsymbol{H} \cdot \hat{\boldsymbol{x}}_k^-$ is weighed more (see Equation 4.6). The same will be true for the reverse case as well.

The matrix $\boldsymbol{Q}$ is part of the predictor equations, in the *a priori* covariance matrix estimate (see Equation 4.4). An increase thereof, will lead to an increase in $\boldsymbol{P}_t^-$.

The effect of different values of $\boldsymbol{R}$ and $\boldsymbol{P}_t^-$ can be seen in more detail in the Appendix Section A.2, and the effect of $\boldsymbol{Q}$ in Section A.5.

It is well known that choosing $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be problematic and that this is highly problem dependent. We remind the reader that the constant matrices $\boldsymbol{R}$ and $\boldsymbol{Q}$ were assigned the values presented in Jaskolski (2017). We could further improve upon our results by fine-tuning these two matrices (for more on the fine-tuning, see Appendix A Section A.5). This, however, is beyond the scope of the current work.

## 4.3 LINEAR REGRESSION MODEL (LRM)

In this section, we present an LRM based approach for predicting linear vessel trajectories. The LRM approach works by estimating $V_t$ using a rolling window linear model. The $V_t$ estimate is used to predict the location of a vessel since the time interval ($\Delta k_t$) and $V_t$ between two subsequent observations is known. The LRM thus differs from the DKF, as the DKF tracks the physical location and speed. In Appendix B, we provide a deeper dive into the LRM and how the Least Squares fit is used to obtain the best fit of the LRM.

We present the LRM approach similarly to the DKF, using two sets of equations - the predictor and measurement update equations (for ease of comparison). The LRM will also only be able to predict in regular time intervals, as determined by $\Delta k_t$.

Let the vector $\boldsymbol{x}_t$ be defined as[§]:

$$\boldsymbol{x}_t = \begin{bmatrix} \lambda_t & \phi_t \end{bmatrix}^T, \tag{4.20}$$

where

- $\lambda_t$ and $\phi_t$, denote the exact LON and the LAT of a vessel, in UTM coordinates at time-step $t$.
- If there is no observation at $t$, we assume $\boldsymbol{x}_t$ to be an all-zero vector.

Let the predictor equations of the LRM be defined by Equations 4.21 and 4.22:

$$\hat{\boldsymbol{x}}_t^- = \hat{\boldsymbol{x}}_{t-1} + \hat{V}_{\omega,t} \cdot \boldsymbol{\Lambda}_t \,, \tag{4.21}$$

$$\hat{V}_{\omega,t} = \nabla \hat{V}_{\omega,t-1} \cdot k_t + \hat{V}_{\omega,c_{t-1}}, \tag{4.22}$$

[§]The symbol $\boldsymbol{x}_t$ is being reused, it should not be confused with the state vector of the DKF.

where

$$\mathbf{\Lambda}_t = \begin{bmatrix} \cos(\psi_t) & \sin(\psi_t) \end{bmatrix}^T . \tag{4.23}$$

The variables of Equations 4.21 - 4.23 are defined below:

- $\hat{\boldsymbol{x}}_t^-$, denotes the predicted position vector using all observations up until time-step $t-1$.
- $\hat{\boldsymbol{x}}_{t-1}$, denotes the updated estimated position vector using all observation up until time-step $t-1$.
- $\omega$, denotes the window size of the LRM (i.e. the number of historic observed observations to take into account).
- $\hat{V}_{\omega,t}$, denotes the vessel's predicted SOG using all observations up until time-step $t-1$, given a window size of $\omega$.
- $\nabla\hat{V}_{\omega,t-1}$, denotes the updated estimated gradient of the LRM using all observations up until time-step $t-1$, given a window size of $\omega$.
- $\hat{V}_{\omega,c_{t-1}}$, denotes the updated estimated $y$-intercept of our LRM using all observations up until time-step $t-1$.
- $\psi_t$, denotes the COG (degrees) of a vessel at time step $t$, $\psi_t$ remains constant until a new COG is recorded from the target vessel.
- $k_t$, denotes the elapsed time in seconds at time-step $t$.

Let the measurement update equations of the LRM be defined by Equations 4.24 - 4.26:

$$\hat{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_t^- + (\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t^-) = \boldsymbol{x}_t \,, \tag{4.24}$$

$$\nabla\hat{V}_{\omega,t} = \frac{\sum_{i=\mathbf{1}_{n_t>\omega}(n_t-\omega)}^{n_t} (V_i - \bar{V}^{\omega})(k_i - \bar{k}^{\omega})}{\sum_{i=\mathbf{1}_{n_t>\omega}(n_t-\omega)}^{n_t} (k_i - \bar{k}^{\omega})} \,, \tag{4.25}$$

and

$$\hat{V}_{\omega,c_t} = \bar{V}^{\omega} - \nabla\hat{V}_{\omega,t} \cdot \bar{k}^{\omega} \,. \tag{4.26}$$

Where,

$$\mathbf{1}_{n_t>\omega} = \begin{cases} 1, & \text{if } n_t > \omega \\ 0, & \text{otherwise} \end{cases} , \tag{4.27}$$

$$\bar{V}^{\omega} = \frac{1}{n_{\omega}} \sum_{i=\mathbf{1}_{n_t>\omega}(n_t-\omega)}^{n_t} V_i \,, \tag{4.28}$$

$$\bar{k}^{\omega} = \frac{1}{n_{\omega}} \sum_{i=\mathbf{1}_{n_t>\omega}(K-\omega)}^{n_t} k_i \,, \tag{4.29}$$

and

$$n_{\omega} = \begin{cases} n_t, & \text{if } n_t < \omega \\ \omega, & \text{otherwise} \end{cases} . \tag{4.30}$$

The variables in Equations 4.24 - 4.30 are defined below**:

- $V_i$, denotes the $i^{\text{th}}$ true SOG observation that was recorded for a particular vessel. In this study, we assumed $V_0 = 4m/s$, meaning we assume that the vessel is already in movement once we start predicting.
- $k_i$, denotes the total time that has elapsed after having recorded the $i^{\text{th}}$ recorded observation.
- $n_t$, denotes the total number of true observations that were recorded after $\Delta k_t \cdot t$ seconds.

The measurement update equations are only updated once a new observation is recorded. If no new observations are recorded, the algorithm will evaluate the predictor equations. The LRM also assumes that the COG remains unchanged until a new observation is observed. Moreover, the predicted position vector at $t-1$ is used to obtain a new estimate of the position vector at $t$.

## 4.4 SINGLE POINT NEIGHBOUR SEARCH (SPNS)

In this section, a complete summary of the method presented by Hexeberg *et al.* (2017) is presented. Note, the variables and symbols used in this section should not be confused with those used throughout the rest of the thesis.

Let,

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 & \boldsymbol{X}_2 & \dots & \boldsymbol{X}_M \end{bmatrix}^T , \tag{4.31}$$

**Note that in this section the subscript $t$ is used as a time-step index, while subscript $i$ is used as an observational index.

be defined as a matrix with all the historic AIS data observations, where $M$ indicates the number of AIS messages recorded.

Moreover let,

$$\boldsymbol{X}_i = \begin{bmatrix} \text{MMSI}_i & t_i & \boldsymbol{p}_i^T & \chi_i & v_i \end{bmatrix}, \tag{4.32}$$

be defined as a vector, where $i$ denote the observation number, $i \in \{1, 2, ..., M\}$. We then define,

- $\text{MMSI}_i$, as the MMSI of $i$,
- $t_i$, the timestamp of $i$,
- $\boldsymbol{p}_i^T$, the position vector of $i$ (at time $t_i$), where $\boldsymbol{p}_i = [\lambda_i, \phi_i,]^T$ which denotes the LON and LAT at $i$, respectively.

A predicted trajectory consists out of $K^s$ predicted positions with different time-instances. At every iteration $k$ a prediction is made, where $k \in \{1, ..., K^s\}$. The predicted state is divided into an *a priori* state $\hat{\boldsymbol{X}}_i^{k-}$ and *a posteriori* state $\hat{\boldsymbol{X}}_i^{k+}$. Let the states be denoted as,

$$\hat{\boldsymbol{X}}_i^{k-} = \begin{bmatrix} \text{MMSI}_i & \hat{t}^k & \hat{\boldsymbol{p}}^k & \hat{\chi}^{k-} & \hat{v}^{k-} \end{bmatrix}, \tag{4.33}$$

and

$$\hat{\boldsymbol{X}}_i^{k+} = \begin{bmatrix} \text{MMSI}_i & \hat{t}^k & \hat{\boldsymbol{p}}^k & \hat{\chi}^{k+} & \hat{v}^{k+} \end{bmatrix}. \tag{4.34}$$

The only difference between Equations 4.33 and 4.34 above, is the COG and SOG. The predicted $\hat{\chi}^{k-}$ and $\hat{v}^{k-}$ in the *a priori* state, represent the predicted COG and SOG between the previous position $\hat{\boldsymbol{p}}^{k-1}$ and $\hat{\boldsymbol{p}}^k$. In the case of the *a posteriori* state, the difference between the current position $\hat{\boldsymbol{p}}^k$ and the next position $\hat{\boldsymbol{p}}^{k+1}$.

The SPNS makes use of a close neighbour (CN) search, where a radius parameter, $r_c$, is defined. Observations within a radius of $r_c$ are queried, given that the observations adhere to a set of predefined constraints. All the close neighbours (CNs) are extracted from historic AIS data. Let the CN set at prediction step $k$ be defined as:

$$C^k = \{\boldsymbol{X}_i \mid d(\hat{\boldsymbol{p}}^k, \boldsymbol{p}_i) \leq r_c,\ \chi_i \in S,\ \boldsymbol{X}_i \in \boldsymbol{X}\} \tag{4.35}$$

where,

- $d(\hat{\boldsymbol{p}}^k, \boldsymbol{p}_i)$ is defined as the Haversine distance (see Equation 3.1) between the LON and LAT of $\hat{\boldsymbol{p}}^k$ and $\boldsymbol{p}_i$.
- $r_c$ is a predefined parameter defined as the search radius in metres, to search for all the CNs within $r_c$ of the current position.
- $S$ is defined as the interval of course angles. Only observations within this interval will be included in the CN set.

Let the course angles $S$ be defined as:

$$S = [\hat{\chi}^{k-} - \Delta\chi,\ \hat{\chi}^{k-} + \Delta\chi] \tag{4.36}$$

where $\Delta\chi > 0$, is a predefined parameter defined as the maximum allowable course deviation.

The above pre-processing steps of the distance and COG deviation filter out all the observations of the CN set, that is not needed for the remaining steps of the SPNS. Let every state that belongs to the set of CNs at prediction step $k$ be denoted as $\boldsymbol{X}_c^k \in C^k$, where $\boldsymbol{X}_c^k = [\text{MMSI}_c^k \ \ t_c^k \ \ \chi_c^k \ \ v_c^k]$ and $c \in \{1, ..., C_n\}$. $C_n$ indicates the number of CNs at $k$.

Let the predicted trajectory $\hat{\boldsymbol{T}}_i$ at state $\boldsymbol{X}_i$ be defined as,

$$\hat{\boldsymbol{T}}_i = \{[\hat{\boldsymbol{p}}^1 \ \ \hat{t}^1],\ [\hat{\boldsymbol{p}}^2 \ \ \hat{t}^2],\ ...,\ [\hat{\boldsymbol{p}}^{K^s} \ \ \hat{t}^{K^s}]\} \tag{4.37}$$

Let the true trajectory $\boldsymbol{T}_i$ given state $\boldsymbol{X}_i$ be defined as,

$$\boldsymbol{T}_i = \{[\boldsymbol{p}^1 \ \ t^1],\ [\boldsymbol{p}^2 \ \ t^2],\ ...,\ [\boldsymbol{p}^L \ \ t^L]\} \tag{4.38}$$

where,

- $K^s$ denote the number of predicted states.
- $L$ denote the number of AIS states recorded.

Note that $K^s$ and $L$ are not necessarily equal, as several prediction steps can be made between two subsequent AIS messages. The first element in $\hat{\boldsymbol{T}}_i$ and $\boldsymbol{T}_i$ are equal, as they are the starting point given by state $\boldsymbol{X}_i$, mathematically we denote this as $\hat{\boldsymbol{T}}_1 = \boldsymbol{T}_1$

### 4.4.1 Position prediction

A new parameter is introduced, $\Delta l$, which denotes the step length from the current observation to the next predicted observation in meters. $\Delta l$ decides how far the next position should be propagated from the current position. Let the predicted position be denoted by,

$$\hat{\boldsymbol{p}}^{k+1} = \boldsymbol{p}_k + \Delta l \cdot [\sin(\hat{\chi}^{k+}) \cdot f(\hat{\phi}^k) \quad \cos(\hat{\chi}^{k+}) \cdot g(\hat{\phi}^k)]^T \,, \tag{4.39}$$

where $f(\hat{\phi}^k)$ and $g(\hat{\phi}^k)$ are functions of the current LAT ($\hat{\phi}^k$), which transforms the LON and LAT from metres to degrees, respectively (Hexeberg *et al.*, 2017). The step length $\Delta l$ reflects the curvature of the sea lanes ahead.

### 4.4.2 Course prediction

The COG value, $\hat{\chi}^{k+}$, is used when calculating the predicted position $\hat{\boldsymbol{p}}^{k+1}$. The *a priori* course $\hat{\chi}^{k+}$ is calculated from the CN set at position $\boldsymbol{p}^k$. Note that the course is periodic in $[0°, 360°]$; therefore, special care must be taken when calculating the CN set's mean COG. Let $\chi_c$ denote a COG value in the CN set. The mean COG of the CN set is calculated as follows:

$$\bar{\chi}_c = \begin{cases} \tan^{-1}(\frac{\bar{s}}{\bar{c}}) & \text{if } \bar{s} > 0,\ \bar{c} > 0 \\ \tan^{-1}(\frac{\bar{s}}{\bar{c}}) + 180° & \text{if } \bar{c} < 0 \\ \tan^{-1}(\frac{\bar{s}}{\bar{c}}) + 360° & \text{if } \bar{s} < 0, \bar{c} > 0 \end{cases}, \tag{4.40}$$

where

$$\bar{s} = \frac{1}{C} \sum_{c=1}^{C} \sin(\chi_c), \tag{4.41}$$

and

$$\bar{c} = \frac{1}{C} \sum_{c=1}^{C} \cos(\chi_c)\,. \tag{4.42}$$

A constant velocity model is used whenever $C^k \notin \Re$ (an empty set). The median course $\tilde{\chi}_c$ can be calculated by calculating $\tilde{s}$ and $\tilde{c}$ instead, it is recommended by Hexeberg *et al.* (2017) to use the median when considering non-linear trajectories, as done throughout this thesis.

### 4.4.3 Speed prediction

The median speed $\tilde{v}_c$ of the CNs set is used to calculate the predicted speed (the speed is inferred from this set). The predicted speed $\tilde{v}_c$ is used to calculate the time passed between the current observation and the predicted observation. Let the time passed be denoted by $\frac{\Delta l}{\hat{v}^{k+}}$. The time update equation is defined as,

$$\hat{t}^{k+1} = \hat{t}^k + \frac{\Delta l}{\hat{v}^{k+}}\,, \tag{4.43}$$

where

- $\hat{t}^k$ denotes the current time.
- $\Delta l$ denotes the distance between the current and the predicted observation (location).
- $\hat{v}^{k+} = \tilde{v}_c$ given the set $C^k$ at $k$.

With this time update, the SPNS predicts in regular distance intervals $\Delta l$, and then calculates the time it took to travel between two subsequent predictions based on the *a priori* speed, calculated from the CN set. The entire SPNS algorithm is presented in Algorithm 4.1.

**Algorithm 4.1** Single Point Neighbour Search Prediction

**Require:** $\boldsymbol{X}_i$, the state predicted from
**Set:**
$\Delta l$, Step length [m]
$r_c$, Search radius [m]
$\Delta\chi$, Maximum course angle deviation [deg]
$K^s$, Number of prediction steps.
$\hat{\boldsymbol{X}}_i^{k-} = \boldsymbol{X}_i$

**for** $k$ **in** $[1, 2, ..., K^s]$ **do**
Find all CNs $\boldsymbol{X}_c^k$ around $\hat{\boldsymbol{X}}_i^{k-}$
Calculate $\hat{\boldsymbol{X}}_i^{k+}$ by:
Calculating $\hat{\chi}^{k+}$ and $\hat{v}^{k+}$ based on the observations in $\boldsymbol{X}_c^k$
Calculate the next predicted position at its predicted point in time:
Calculate $\hat{\boldsymbol{p}}^{k+1}$ with Equation 4.39
Calculate $\hat{t}^{k+1} = \hat{t}^k + \frac{\Delta l}{\hat{v}^{k+}}$
**Set:** $\hat{\boldsymbol{X}}_i^{(k+1)-} = [\text{MMSI}_i \;\; \hat{\boldsymbol{p}}^{k+1} \;\; \hat{\chi}^{k+} \;\; \hat{v}^{k+}]$
**end for**

The set of hyperparameters we used in this study is listed in Table 4.1. They are the same as those used within the original study of Hexeberg *et al.* (2017), ideal for curved trajectory prediction.

| Decision Parameter | Value | Explanation |
|---|---|---|
| $r_c$ | 50m | Search radius for the CNs |
| $\Delta l$ | $2r_c$ | Prediction step length [m] |
| $\Delta\chi$ | 25° | Maximum course deviation |
| $\hat{\chi}_i^{k+}$ | $\tilde{\chi}_c$ | Course prediction used at every iteration $k$ |
| $\hat{v}_i^{k+}$ | $\tilde{v}_c$ | Speed prediction used at every iteration $k$ |

Table 4.1: Curved Trajectory Prediction Decision parameters of the SPNS

### 4.4.4 SPNS query setup

Since the SPNS algorithm requires querying historic observations within a specified radius, all the data was loaded into a PostgreSQL database (PostgreSQL, 2021). An extension was added to PostgreSQL called PostGIS, which allows for improved spatial queries with a datatype called geometry (PostGIS, 2021). The PostGIS plugin uses a unique kind of indexing. Querying observations in PostgreSQL with PostGIS allows for vessels within a given radius from the search point to be extracted (i.e. $\boldsymbol{C}^k$ can be constructed).

## 4.5 LINEAR REGRESSION MODEL WITH *A PRIORI* COG INFORMATION (LRMAC)

In this section, we present a novel algorithm called the LRMAC, as mentioned before, this method uses historic (*a priori*) AIS data to predict vessel trajectories. The LRMAC is an extension of the LRM, allowing for the prediction of non-linear trajectories (as shown in Figure 3.12). Spatial Maps (SMs) are used as *a priori* information (introduced in Section 3.2). First, we discuss some unit conversions that the algorithm requires and then introduce the LRMAC.

### 4.5.1 LRMAC unit conversions

The LRMAC does not require the conversion of the LAT, LON coordinates as the non *a priori* methods do. However, the LRMAC converts the SOG from m/s to degrees/s, working in the LAT and LON degrees domain. This allows the LRMAC, more specifically the underlying LRM, to predict a vessel's displacement in terms of degrees LAT and LON for each unit of time.

Assuming that the SOG ($V_t$) is in $m/s$ (as converted by Equation 4.1), the LRMAC requires an extra conversion of the SOG in order for it to work as intended. Equation 4.44, shows the conversion of the SOG from m/s to degrees/s ($^\circ/s$).

$$V_t^{''} = \frac{V_t}{\bar{l}} \tag{4.44}$$

with:

$$\bar{l} = \frac{2\pi}{360} \times 6378000 = 111137m \tag{4.45}$$

The constant $\bar{l}$ can be interpreted as the average number of metres that one degree of LAT and LON span on Earth. It is assumed for the remainder of the thesis that when we refer to the SOG, with respect to the LRMAC, its unit is $^\circ$/s, reusing the symbol $V_t$.

Since LRMAC is an extension of the LRM the same assumptions hold, except for the constant COG assumption. The LRMAC also predicts in regular time intervals ($\Delta k_t = 1$s), and a constant velocity is assumed based on LRM fit. The velocity will be updated, once an update is received from the target vessel.

### 4.5.2 The proposed method

The proposed method uses SMs (as *a priori* information) to update the COG, improving prediction accuracy and allowing for non-linear trajectory prediction. Programmatically speaking, SMs can easily be loaded into memory, and their sizes are relatively small given the information they contain. The symbols in this section continue from those used in Section 4.3, due to the LRMAC being an extension of the LRM.

The COG value is dynamically updated using three SM matrices: $\boldsymbol{K}$, $\boldsymbol{\Psi}$, and $\boldsymbol{\Sigma}$. The COG is used to determine the SOG in the respective latitudinal, and longitudinal directions (see Equation 4.21), and the update thereof allows for the non-linear trajectory prediction.

The predictor equations of the LRM are modified to obtain the LRMAC. The COG value is calculated whilst predicting, computed at step $t$. However, it is only applied during step $t+1$ when the LRM predicts the next location as the COG is used in calculating the displacement of a vessel in the respective latitudinal and longitudinal directions.

Let $\hat{\boldsymbol{x}}_t^-$ be the predicted position vector as in Equation 4.21. Let

$$\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}} = [n_{\hat{\phi}}, n_{\hat{\lambda}}] \tag{4.46}$$

denote the SM index positions associated with $\hat{\boldsymbol{x}}_t^- = [\hat{\lambda},\, \hat{\phi}]^T$. In other words with $[n_{\hat{\phi}},\, n_{\hat{\lambda}}]$ we can extract the values in $\boldsymbol{K}$, $\boldsymbol{\Psi}$, and $\boldsymbol{\Sigma}$ associated with $(\hat{\phi},\, \hat{\lambda})^\dagger$, by making use of matrix subscripting[‡].

Let us construct an index matrix $\boldsymbol{H}$:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{n}_{\hat{\phi}-\eta\cdot\kappa,\,\hat{\lambda}-\eta\cdot\kappa} & \cdots & \boldsymbol{n}_{\hat{\phi}-\eta\cdot\kappa,\,\hat{\lambda}} & \cdots & \boldsymbol{n}_{\hat{\phi}-\eta\cdot\kappa,\,\hat{\lambda}+\eta\cdot\kappa} \\ \vdots & \ddots & \vdots & \iddots & \vdots \\ \boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}-\eta\cdot\kappa} & \cdots & \boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}} & \cdots & \boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}+\eta\cdot\kappa} \\ \vdots & \iddots & \vdots & \ddots & \vdots \\ \boldsymbol{n}_{\hat{\phi}+\eta\cdot\kappa,\,\hat{\lambda}-\eta\cdot\kappa} & \cdots & \boldsymbol{n}_{\hat{\phi}+\eta\cdot\kappa,\,\hat{\lambda}} & \cdots & \boldsymbol{n}_{\hat{\phi}+\eta\cdot\kappa,\,\hat{\lambda}+\eta\cdot\kappa} \end{bmatrix} \tag{4.47}$$

where,

- $\eta$ denotes the neighbourhood parameter, and
- $\kappa$ represents the width and length of a SM cell, as shown on the axes of Figure 3.8.

The index matrix $\boldsymbol{H}$ is used to select a specific sub-grid/matrix subset from $\boldsymbol{K}$ and $\boldsymbol{\Sigma}$, at the index locations contained in $\boldsymbol{H}$.

Let the *a priori* cell counts, of the area surrounding $\hat{\boldsymbol{x}}_t^-$, be denoted by

$$\boldsymbol{K}_H, \text{ where } \boldsymbol{K}_H \subset \boldsymbol{K}. \tag{4.48}$$

The size of the aforementioned area is determined by $\eta$. Moreover, let the *a priori* average COG associated with $\hat{\boldsymbol{x}}_t^-$, be denoted by,

$$\boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}}, \text{ where } \boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}} \in \boldsymbol{\Psi}. \tag{4.49}$$

[†]The brackets of $(\hat{\phi},\, \hat{\lambda})$ indicate that they are coordinates.

[‡]Matrix subscripting refers to extracting values at the corresponding index locations in the associated matrices, similar to matrix slicing in NumPy (Harris *et al.*, 2020).

Lastly, let the COG SD associated with $\hat{\boldsymbol{x}}_t^-$, be defined as,

$$\boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}}, \text{ where } \boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}} \in \boldsymbol{\Sigma}. \tag{4.50}$$

### 4.5.3 Updating the COG using *a priori* information

We first need to calculate the confidence we have in the *a priori* COG value $\boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}}$. This confidence measure ($\rho$) ranges between zero and one ([0, 1]), and is used to scale the contribution of the *a priori* COG information. It is determined by the number of observations in the neighbourhood, as shown by Equation 4.51.

The scaling factor allows us to determine how much of the *a priori* information at the predicted cell should contribute to the COG update. If the *a priori* count for the current cell (in SM $\boldsymbol{K}$) is higher compared to the other cells in the neighbourhood, we can more confidently say that the predicted location is in an area where historically, many vessels have travelled before. If the current cell count is lower compared to the surrounding cells, the *a priori* COG information will contribute less, assigning more weight to the previously used COG. This, in effect, allows a vessel to stay in the highways. The confidence factor is calculated as follows:

$$\rho = \mathbf{1}_{\boldsymbol{\Psi}} \cdot \frac{\boldsymbol{K}_{\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}}}{\max(\boldsymbol{K}_{\boldsymbol{H}})}, \tag{4.51}$$

where

- $\rho$ denotes the confidence (scaling factor) that we have in our prediction as determined by the current predicted position of a vessel, $\hat{\boldsymbol{x}}_t^-$.
- $\boldsymbol{K}_{\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}}$ denotes the cell count value associated with $(\hat{\phi},\, \hat{\lambda})$. The index $\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}$ is used to extract the count from $\boldsymbol{K}$.
- $\mathbf{1}_{\boldsymbol{\Psi}}$ denotes the indicator function that sets $\rho$ to zero. The indicator function enforces the restrictions that we impose on whether the COG should be updated or not.

If $\rho$ is close to one, we can be confident in the *a priori* value $\boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\,\hat{\lambda}}}$, and if it is close to zero, the opposite as seen in Equation 4.53. Two factors influence $\rho$:

1. If few vessels have traversed the cell associated with $\boldsymbol{x}_t$ relative to its neighbouring cells,

$\frac{\boldsymbol{K}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}}{\max(\boldsymbol{K}_{\boldsymbol{H}})}$ would be relatively close to zero, indicating that the *a priori* COG information should have less impact in updating $\boldsymbol{\Psi}_{t+1}$. The same is true in the opposite case, values closer to one would have a more significant contribution to updating $\boldsymbol{\Psi}_{t+1}$.

2. The value of the indicator function in Equation 4.52. Let

$$\mathbf{1}_{\boldsymbol{\Psi}} = \begin{cases} 0, & \text{if } \boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}} \notin \Re \\ 0, & \text{if } \boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}} \notin \Re \\ 0, & \text{if } \boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}} > 10^{\circ} \\ 0, & \text{if } \max(\boldsymbol{K}_{\boldsymbol{H}}) = 0 \\ 1, & \text{otherwise} \end{cases}, \tag{4.52}$$

where $\mathbf{1}_{\boldsymbol{\Psi}}$ evaluate to zero if:

- $\boldsymbol{\Psi}$ or $\boldsymbol{\Sigma}$ contains no information at index $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ (i.e. $\notin \Re$). This implies that there is no *a priori* information available for us to make use of.
- The COG SD at $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ is larger than 10° [§]. This implies that many vessels have traversed through the cell associated with $\hat{\boldsymbol{x}}_t^-$, all going in different directions. Implying the *a priori* COG value is less reliable.

The COG value that will be used at the next iteration can now be updated as follows:

$$\hat{\psi}_{t+1} = (1-\rho)\psi_t + \rho\boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}, \tag{4.53}$$

where

- $(1-\rho)$ indicates the role that the previous observed COG should have in the COG update.
- $\psi_t$ denotes the previously observed or predicted COG. An observed COG will always receive preference over a predicted COG, as the COG value is updated in the measurement update equations.
- $\boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$ denotes the *a priori* COG scalar value at index $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ of the current predicted $\hat{\boldsymbol{x}}_t^-$.

[§]An SD of 10° was chosen as our threshold as it yielded good results, the optimisation of this threshold was deemed out of scope for the thesis and is considered as future work.

The LRMAC calculates the COG in the current step $t$, but it is only used in the next step $t+1$ of the predictor equations. The constant COG in Equation 4.23 is replaced with the COG value calculated in Equation 4.53. Let Equation 4.54 replace Equation 4.23 for the LRMAC, with the updated COG.

$$\mathbf{\Lambda}_t = \begin{bmatrix} \cos(\hat{\psi}_t) & \sin(\hat{\psi}_t) \end{bmatrix}^T, \tag{4.54}$$

where $\hat{\psi}_t$, refers to the *a priori* COG calculated at the previous iteration. Note that $\hat{\psi}_{t+1}$ at the previous time-step $(t-1)$, is equal to $\hat{\psi}_t$ at the current time-step $(t)$ if no new COG value is observed.

The LRMAC in effect adds two additional equations to the LRM. The first equation added, is a predictor equation (Equation 4.53) which allows for the COG to be updated by *a priori* information. The second equation is an alteration of an existing equation. Equation 4.23 is replaced by Equation 4.54, when no new COG value is observed. Equation 4.23 is used otherwise.

### 4.5.4 A flow diagram representation of the LRMAC

In Figure 4.2, a flow diagram of the LRMAC methodology is depicted. The diagram shows the LRM and how *a priori* course (COG) information is added to extend the LRM into the LRMAC. Initialisations are in green, functions are in blue, the predicted location is in grey and parameter extracts are denoted in orange. It is assumed that all pre-processing has already been applied to the dataset.

The first step is to construct the SMs from the dataset containing all historic AIS data and initialise all parameters (see Section 3.2). The last $\omega$ recorded observations are used as additional input parameters in the LRM, specifically for the measurement update equations. The LRM consists of two sets of equations, measurement update and predictor equations. The measurement equations are used to update the predictor parameters and the predictor equations to predict the next set of coordinates.

The algorithm starts at the indicated red dot in the flow diagram. The LRM is used to estimate the LAT and LON at the next time step. All predictions are made at regular spaced time intervals $\Delta k_t$. The LRM assumes a constant SOG and COG, based on the last $\omega$ observations.

Given the predicted LAT and LON, we extend the LRM into the LRMAC by dropping the constant

COG assumption. The COG will now be updated based on *a priori* COG information at the location predicted by the LRM. The updated COG will only be incorporated at the next iteration when the LRM predicts the new location. The COG is used to calculate the SOG in the respective LAT and LON directions. The extension of the LRM allows the predictions to follow historic movement trends in the SMs, allowing for non-linear trajectory prediction.

The COG update is done as follows:

1. Given the predicted location $\hat{\boldsymbol{x}}_t^-$, the corresponding index $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ in the SMs can be calculated, and the neighbouring indexes matrix $\boldsymbol{H}$.
2. Using $\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}$ and $\boldsymbol{H}$, the *a priori* values: $\boldsymbol{K}_{\boldsymbol{H}}$, $\boldsymbol{\Psi}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$, $\boldsymbol{\Sigma}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$, and $\boldsymbol{K}_{\boldsymbol{n}_{\hat{\phi},\hat{\lambda}}}$ is extracted and used in Equation 4.51 and Equation 4.52 to calculate the updated COG value ($\hat{\psi}_{t+1}$).

The LRMAC allows the COG to be updated dynamically based on historic AIS data, allowing the LRMAC to follow historic movement patterns of vessels. All other parameters, including the COG, will update once new observations are received from the vessel, updating the measurement equations.

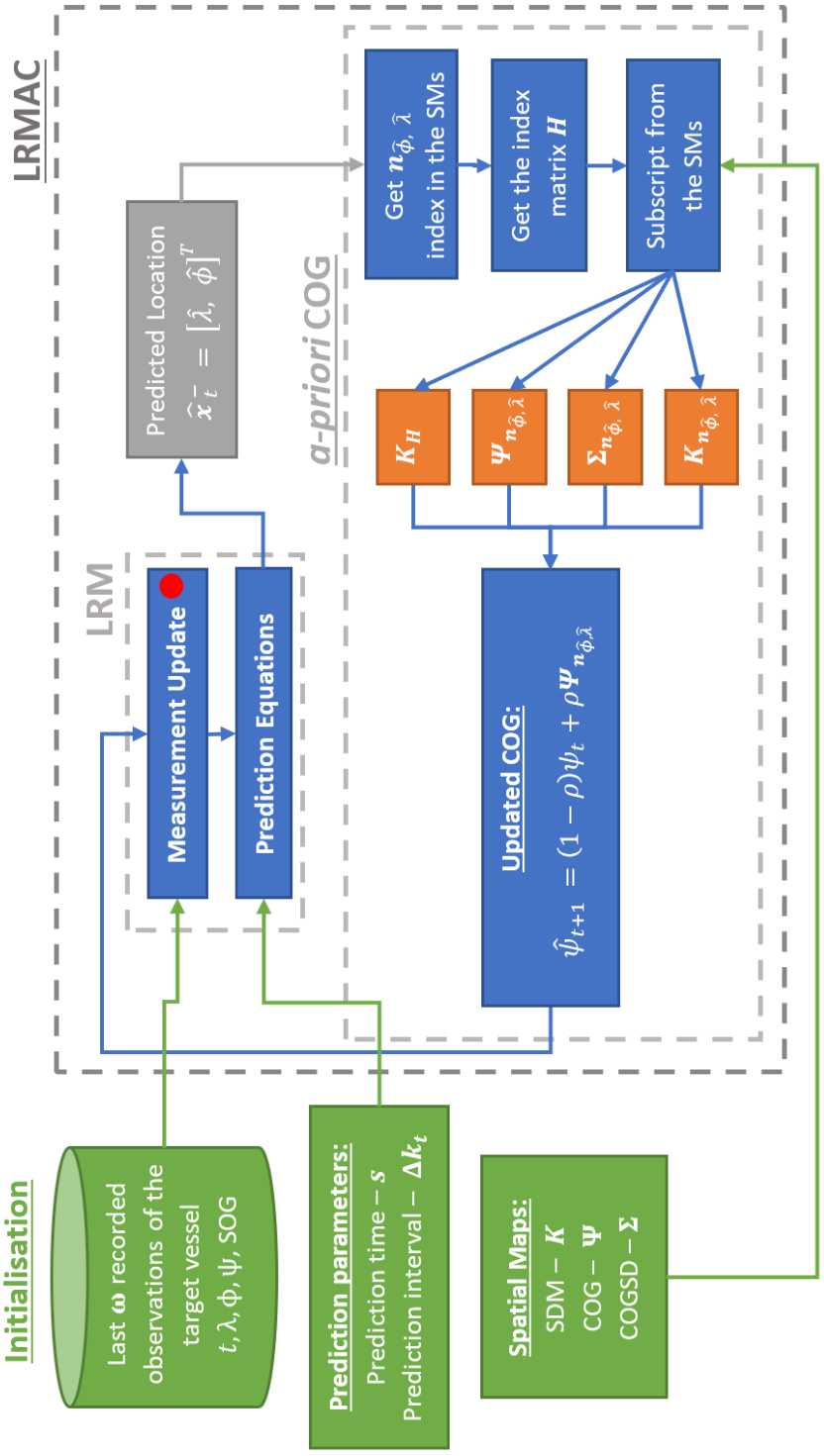


Figure 4.2: LRMAC flow diagram of extending the LRM into the LRMAC.

## 4.6 SUMMARY

In this chapter, we discussed all the methods implemented in this thesis, both non *a priori* and *a priori* methods. The necessary unit conversions that these methods require are also discussed. First, we discussed the non *a priori* methods which include the DKF and the LRM, followed by the *a priori* methods, the SPNS and the LRMAC. A flow diagram is also presented to show how the LRM was extended into the LRMAC (see Figure 4.2). Each method is discussed in detail and should enable the reader to understand the inner workings of each. In the next chapter, a comparison between the two non *a priori* methods is made, followed by a comparison between the two *a priori* methods.

# CHAPTER 5

# RESULTS

In this chapter, we compare all the methods introduced in Chapter 4. First, we discuss the trajectory subsets the methods were compared on. In Section 5.2, we present the results of a comparison study between the DKF and the LRM. In Section 5.3, we present the results obtained after having compared the LRMAC and the SPNS algorithm with each other. These two sections follow a similar layout; the experimental setup is first discussed, then the trajectories considered, and they end with empirical results. We end the chapter by comparing the algorithmic complexities between all the methods.

## 5.1 TRAJECTORY TEST SETS

The sets of unique vessel trajectories the methods were tested on (consisting of Cargo and Tanker vessel trajectories) can be found in Table C.1 and C.2 in Appendix C. The trajectories in both the aforementioned tables were extracted from the dataset which was published by (Ray *et al.*, 2019), after the cleaning steps in Section 3.1 were applied.

The non *a priori* methods were tested on 30 linear vessel trajectories, while the *a priori* methods were tested on 40 non-linear vessel trajectories. These trajectories were similar to those depicted in Figure 3.12.

Vessel trajectory subsampling methods were implemented for all of the methods. This effectively increases the number of trajectories the methods were tested on. The subsampling methods are discussed in Section 5.2.1.1 and 5.3.1.2, respectively.

The extracted trajectories for the *a priori* methods are vessels from both highways depicted in Figure 3.9, consisting of Cargo and Tanker vessels. These areas provide us with many curved trajectory examples, which we require to compare LRM, LRMAC and SPNS with one another.

## 5.2 NON *A PRIORI* METHODS

In this section we compare the LRM (see Section 4.3) to the DKF (see Section 4.2). We discuss the experimental design for testing the methods, followed by the comparison between the two methods.

### 5.2.1 Experimental Design

In this section, we discuss how we generated multiple trajectory subsets to test the methods on, hyperparameter selection, and how we measured prediction accuracy.

#### *5.2.1.1 Trajectory subsampling method*

Consider linear trajectories similar to the linear trajectory in Figure 3.12, but more specifically those in Table C.1. Below we present a trajectory subsampling approach that will allow us to increase the number of trajectories on which we can test the performance of the LRM and the DKF. The method will create undersampled trajectories that span different time lengths. The main aim of this augmentation is to make it possible to evaluate the performance of the LRM and the DKF for different prediction horizons and trajectory sparsity levels.

Let the time-intervals (prediction horizons) that are considered be represented by the set:

$$\boldsymbol{\delta}_s = \{240\text{s},\ 300\text{s},\ 360\text{s},\ \cdots,\ 1800\text{s}\}, \tag{5.1}$$

and the undersampling rate (observational step-sizes) be represented by the set:

$$\boldsymbol{\lambda}_s = \{2,\ 3,\ 4,\ ...,\ 21\}. \tag{5.2}$$

Moreover, let $n_\delta = |\boldsymbol{\delta}_s|$ and $n_\lambda = |\boldsymbol{\lambda}_s|$ denote the number of time intervals and step-sizes, respectively.

Let $\boldsymbol{T}$ denote the list of trajectories in Table C.1. Furthermore, let $\boldsymbol{T_i}$ denote the $i^{\text{th}}$ trajectory in $\boldsymbol{T}$. Moreover, let $\mathcal{C}_{\boldsymbol{T}_i}$ denote the set that is constructed by undersampling trajectory $\boldsymbol{T_i}$ using all possible combinations of time-intervals and step-sizes contained within $\boldsymbol{\delta}_s$ and $\boldsymbol{\lambda}_s$. Lastly, let $\mathcal{C}$ be the set that contains all possible $\mathcal{C}_{\boldsymbol{T}_i}$. The algorithm for constructing $\mathcal{C}$ is presented in Algorithm

5.1. Algorithm 5.1 makes use of Algorithm 5.2. Algorithm 5.2 describes how a trajectory is undersampled given an interval size $\delta_{s_i}$ and step size $\lambda_{s_i}$. Algorithm 5.2 generates multiple subsets of trajectories which are undersampled by slicing up a trajectory up into different interval lengths ($\delta_{s_i}$), and undersampling within each sub sample ($\lambda_{s_i}$). Subsampling, mimics vessels that enter weak AIS reception areas or that send minimal status updates. This method of subsampling allows us to compare the performance of the LRM with the DKF with respect to different scenarios.

**Algorithm 5.1** Generating $\mathcal{C}$

**Require:** $\boldsymbol{T}$, $\boldsymbol{\delta}_s$, $\boldsymbol{\lambda}_s$, $\mathcal{C}$
  $\mathcal{C} = \emptyset$ ▷ Where $\emptyset$ represents an empty set
  **for** $\boldsymbol{T}_i$ **in** $\boldsymbol{T}$ **do**
    $\mathcal{C}_{\boldsymbol{T}_i} = \emptyset$
    **for** $\delta_{s_i}$ **in** $\boldsymbol{\delta}_s$ **do**
      **for** $\lambda_{s_i}$ **in** $\boldsymbol{\lambda}_s$ **do**
        $\mathcal{C}_{\boldsymbol{T}_i}$.append(*SubSampleTraj*($\boldsymbol{T}_i$, $\delta_{s_i}$, $\lambda_{s_i}$)) ▷ As in Algorithm 5.2
      **end for**
    **end for**
    $\mathcal{C}$.append($\mathcal{C}_{\boldsymbol{T}_i}$)
  **end for**

**Algorithm 5.2** *SubSampleTraj()* - generating an undersampled trajectory of $\boldsymbol{T}_i$

**Require:** $\boldsymbol{T}_i$ , $\delta_{s_i}$, $\lambda_{s_i}$
  $n = |\boldsymbol{T}_i|$ ▷ The observational length in seconds of $\boldsymbol{T}_i$
  $\iota = 0$ ▷ Interval starting point (s)
  $\boldsymbol{\alpha} = \emptyset$
  **seq** = sequence(start = 0, end = $\delta_{s_i}$, step = $\lambda_{s_i}$) ▷ Index array to perform undersampling

  **while** $\iota \leq$ n **do**
    $\boldsymbol{T}_{i_{\text{interval}}}$ = extract($\boldsymbol{T}_i$, $\iota$, $(\iota + \delta_{s_i})$) ▷ Extract a new observational subset from $\boldsymbol{T}_i$
    $\boldsymbol{T}_{i_{\text{Undersample}}} = \boldsymbol{T}_{i_{\text{interval}}}$[**seq**] ▷ Extracting undersampled trajectory from $\boldsymbol{T}_{i_{\text{interval}}}$
    $\boldsymbol{\alpha}$.append($\boldsymbol{T}_{i_{\text{Undersample}}}$) ▷ Appending subset from $\boldsymbol{T}_{i_{\text{interval}}}$ according to the $\lambda_{s_i}$ to the set
    $\iota = \iota + \delta_{s_i}$ ▷ Update the starting point for next iteration
    **if** $(\iota + \delta_{s_i}) > n$ **then** ▷ Check to see if we can generate another undersampled subset
      **return**($\boldsymbol{\alpha}$)
    **end if**
  **end while**

The reasoning behind the constant time intervals for subset generations is because AIS transponders transmit messages at regular time intervals (Jaskolski, 2017). However, the data is not always sent to receivers on time, and signals can be interrupted. We chose to use constant time intervals to subdivide our trajectories as it is more realistic, than constant displacement intervals. The reason

is that constant displacement intervals would result in nonsensical conclusions being drawn because the velocities of the vessels differ. Keeping time a constant, allows us to evaluate the prediction performance on similar time frames due to the constant AIS time intervals.

#### *5.2.1.2 DKF Hyperparameters*

The $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices can be thought of as the hyperparameters for the DKF. The reason being, that the DKF's performance is dependent on the initialisation of these matrices.

How $\boldsymbol{Q}$, $\boldsymbol{R}$ and $\boldsymbol{P}_0$ were initialised are discussed in Section 4.2.1.1.

#### *5.2.1.3 LRM Hyperparameters*

The LRM only has one hyperparameter, the window size ($\omega$). Throughout all the experiments of the LRM the window size was set to three, $\omega = 3$. This ensures that the results that are reported remain consistent. The value, however, was determined via experimentation (see Section 5.3.1.4).

#### *5.2.1.4 Prediction Accuracy Measurement*

Both the DKF and the LRM are applied to every trajectory in every undersampled set $\mathcal{C}_{T_i}$, generated by Algorithm 5.1. For each method and undersampling rate, we calculated the Mean Euclidean Distance (MED) as follows:

$$\text{MED}_{\boldsymbol{T}_i,\lambda_{s_i}} = \frac{1}{n \cdot n_\delta} \sum_{j=1}^{n_\delta} \sum_{k=1}^{n} \sqrt{(x_{\boldsymbol{T}_i,k} - \hat{x}_{\boldsymbol{T}_i,k,\delta_{s_j},\lambda_{s_i}})^2 + (y_{\boldsymbol{T}_i,k} - \hat{y}_{\boldsymbol{T}_i,k,\delta_{s_j},\lambda_{s_i}})^2} \,, \tag{5.3}$$

and

$$\text{MED}_{\lambda_{s_i}} = \frac{1}{|\boldsymbol{T}|} \sum_{\boldsymbol{T}_i \in \boldsymbol{T}} \text{MED}_{\boldsymbol{T}_i,\lambda_{s_i}}. \tag{5.4}$$

The variables in the above two equations are defined below:

- $n$, denotes the length of the original trajectory.
- $x$ and $y$, denote the original trajectory observations (LON, LAT).
- $\hat{x}$ and $\hat{y}$, denote the estimated $x$- and $y$-coordinate values of trajectory $\boldsymbol{T}_i$.
- $\boldsymbol{T}_i$, denotes trajectory $i$ in Table C.1.

- $\text{MED}_{\lambda_{s_i}}$, denotes the average MED over all the trajectories for every undersample rate in $\boldsymbol{\lambda}_s$.

### 5.2.2 LRM and DKF Comparison

In this section, we compare the overall performance of the LRM and the DKF. We compare the prediction accuracy of both methods for each undersampling rate. A larger undersampling rate translates into more sparse trajectories. The purpose of this comparison is to see if there are any significant differences in the prediction accuracy of the LRM and the DKF.

Figure 5.1 depicts the computed $\text{MED}_{\lambda_{s_i}}$ values (as calculated by Equation 5.3) and the corresponding standard deviations. The error at each $\lambda_{s_i}$ is the mean error for all time frames $\boldsymbol{\delta}_s$. This is calculated as the difference between the predicted coordinate and the observed coordinate.

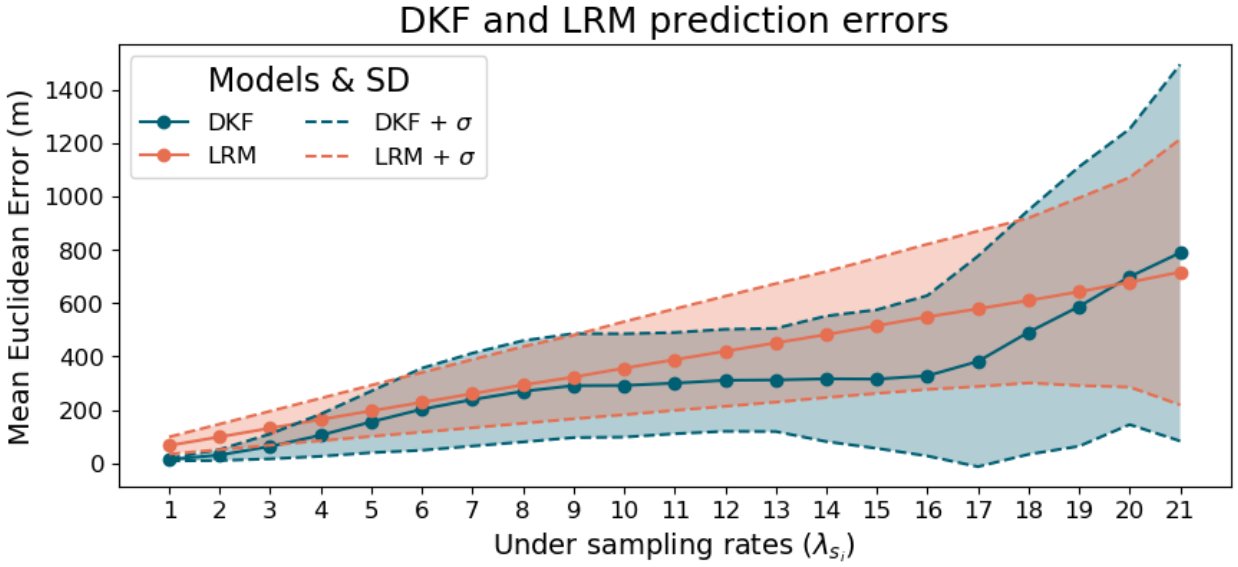


Figure 5.1: $\text{MED}_{\lambda_{s_i}}$ values for the DKF and the LRM methods in metres.

First, let us put the errors shown in Figure 5.1 in perspective. The physical length of Cargo vessels ranges from 137m to 400m (Rodrigue *et al.*, 2016), and for Tanker vessels from 205m to 405m (Notteboom *et al.*, 2020) (see Figure 3.13 to put the size of one of these vessels in perspective). That being said, when inspecting the errors of both the DKF and the LRM in Figure 5.1, we see no significant difference in prediction error between the two methods. We also see that the confidence interval of the LRM and the DKF encapsulates each other's MED values.

As the undersampling rate increases, the LRM's error and standard deviation increase at an almost

constant rate, while the error of the DKF and its associated standard deviation bands increases non-linearly. When looking at the MED of the DKF and the LRM between $\lambda_{s_i} = 13$ and $\lambda_{s_i} = 17$ it looks as if the DKF outperforms the LRM. However, a prediction difference of 200m with respect to Cargo and Tanker vessels, is the difference between a vessel's bow or stern being in the predicted location (due to their size).

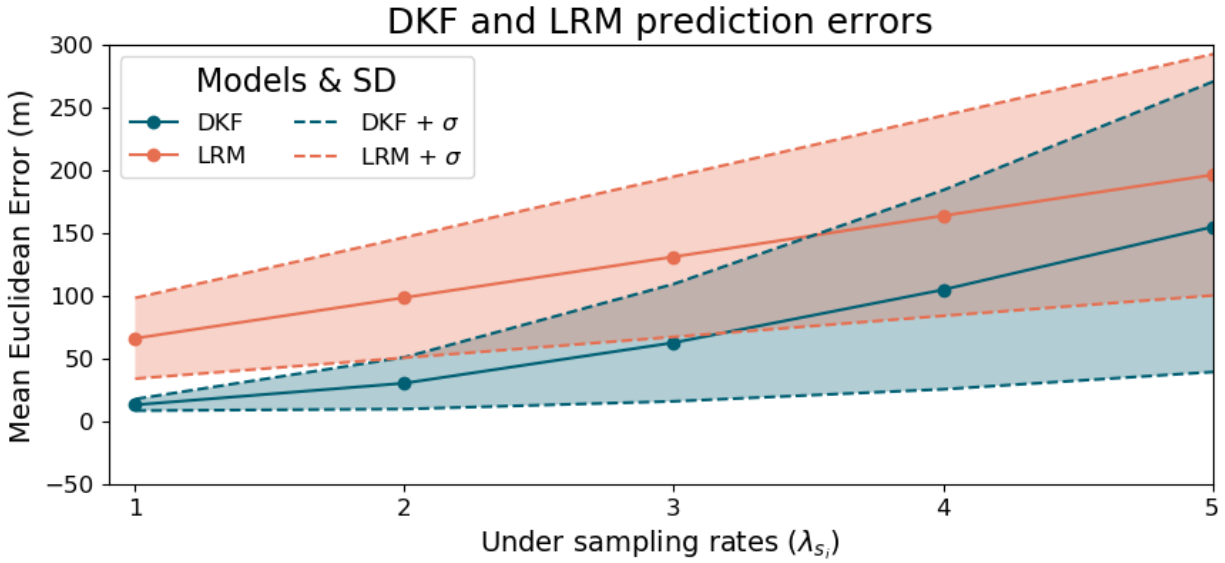


Figure 5.2: Zoomed-in undersampling rates of Figure 5.1.

Figure 5.2 depicts a sub-region of Figure 5.1 (the region associated with undersampling rates $\{2, 3, 4, 5\}$). Even though it visually looks as if the DKF outperforms the LRM, the difference is insignificant due to the size of Cargo and Tanker vessels.

However, the most important conclusion is that the DKF does not perform significantly better than the LRM on near-linear trajectories, given the size of vessels and that the confidence intervals of both methods almost always overlap. The DKF is also more complex to implement compared to the LRM.

### 5.2.3 Case Study - A Comparison between the LRM and DKF

Given the overall results we now investigate a specific case study. We showcase an example where the LRM outperforms the DKF in terms of prediction accuracy (does not happen often). The aim behind this section is to enable the reader to better understand the overall results presented in

Section 5.2.2.

The vessel with MMSI 304927000 and sub-trajectory where $\lambda_{s_i} = 3$ was used for this case study, summarised in Table C.1.

#### *5.2.3.1 Model ability to predict vessel SOG*

First, we remind the reader that both methods update their estimates as new observations are observed from a vessel. In Figure 5.3, a plot is shown of the SOG associated with MMSI 304927000 for each recorded observation of the trajectory in question. Note that the time intervals between each observation are not regular. We observe an average speed pf 10.39 kt and a median speed of 10.5 kt, and a short period of rapid decrease in speed after observation 429. Note when we refer to the speeds in the longitudinal and latitudinal directions here, we actually mean velocity, i.e. the measured values we report, can become negative.

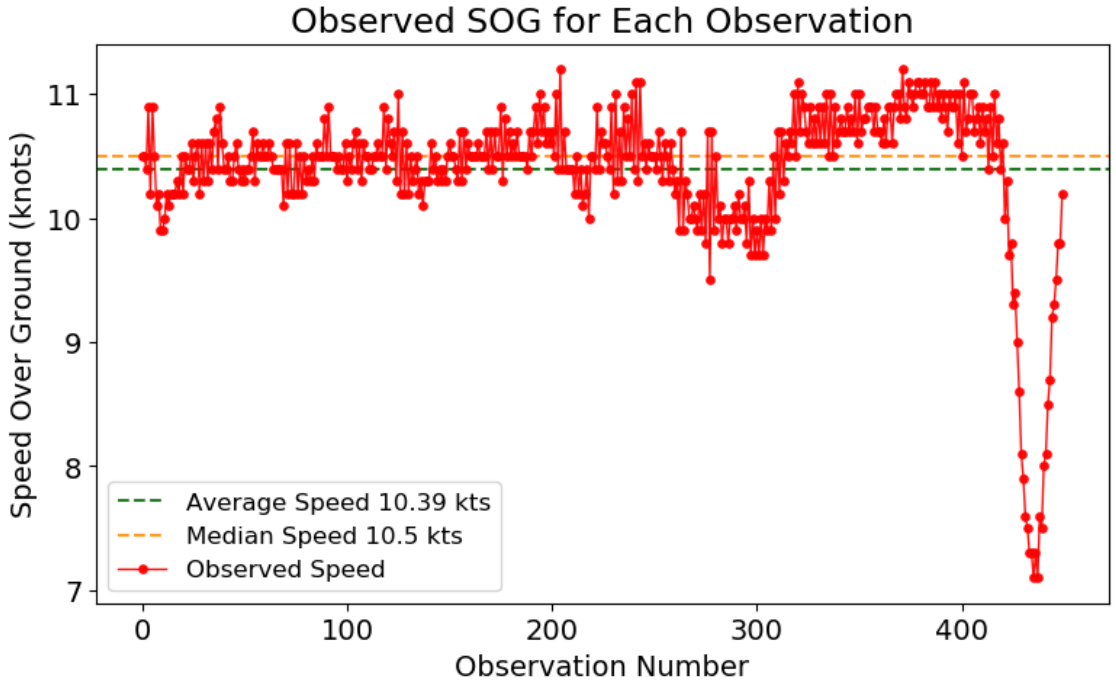


Figure 5.3: SOG per observation MMSI - 304927000 as denoted in Table C.1

Figure 5.4, depicts the latitudinal speed predicted by the LRM. We observe that the LRM (blue) is able to closely track the historical observed speeds (in red). The LRM is able to provide a good estimate of the SOG. When looking at the DKF's estimation in Figure 5.5, we see that the predictions are far more erratic compared to the LRM. The erratic SOG predictions of the DKF are most probably due to matrices $\boldsymbol{P}_t^-$ and $\boldsymbol{K}_t$ not being able to converge (reach a form of stability) in the presence of noisy observations (see Section 4.2.1.1).

Looking at the prediction performance of the DKF in Figure 5.5, it looks as if both of the aforementioned matrices are overcorrecting for the errors made once an update is received from the target vessel. As new observations are received from the vessel, the DKF updates through the measurement updates. We even note that the DKF predicts a negative SOG value (i.e. the vessel is moving backwards or in the complete opposite direction than it was moving in at first). This negative SOG occurs due to DKF trying to correct an off-course prediction after receiving an actual observation.

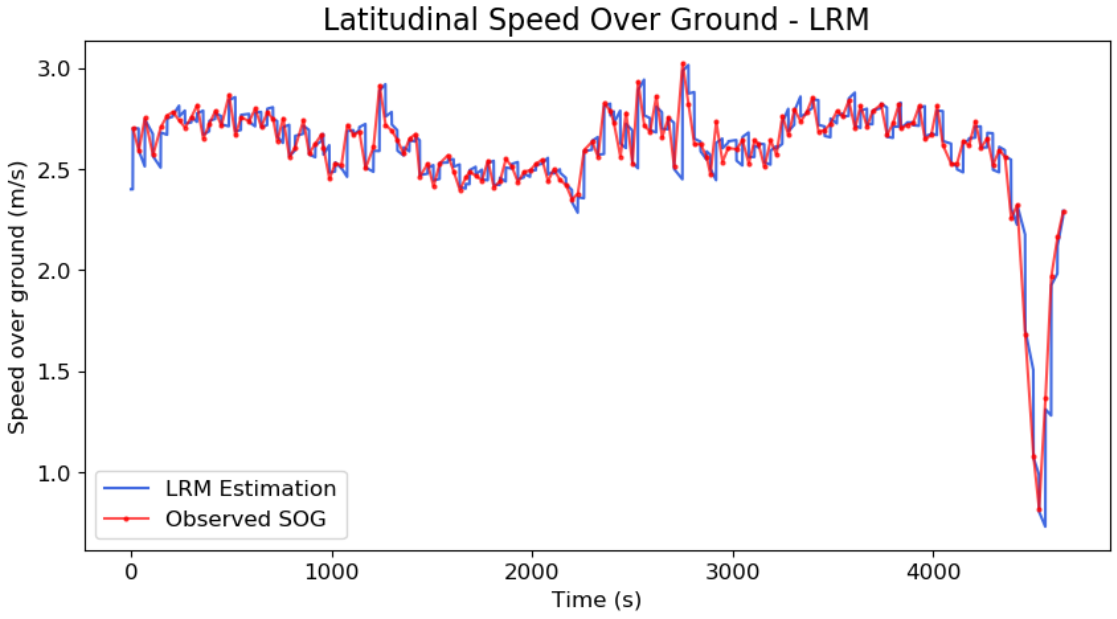


Figure 5.4: Latitudinal speed prediction by the LRM of MMSI 304927000, $\lambda_{s_i} = 3$.

When looking at Figure 5.5 above, one might assume that the erratic prediction of the SOG that the DKF made will result in an erratic coordinate prediction. This, however, is not always the case. The DKF jointly predicts the SOG and the vessel's coordinates, whereas the LRM predicts

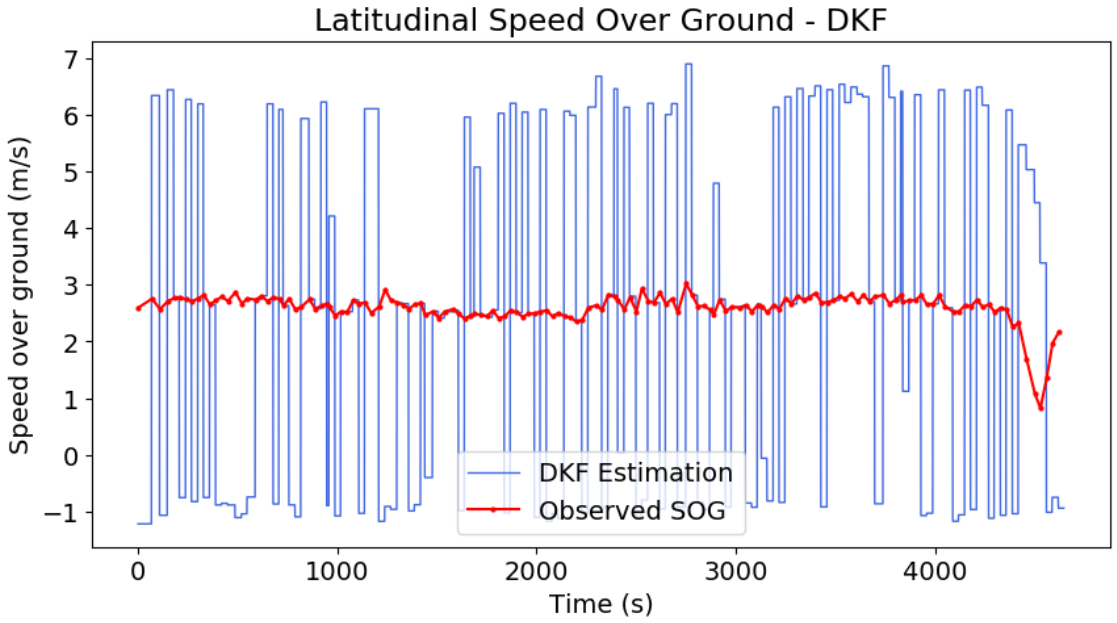


Figure 5.5: Latitudinal speed (velocity) prediction by the DKF of MMSI 304927000, $\lambda_{s_i} = 3$.

the SOG and derives the coordinates from the predicted speed as the time elapsed is known.

Note that the $x$-axis denotes the observation number in Figure 5.3, and in Figures 5.4 and 5.5 the $x$-axis denotes elapsed time (the observations are not regularly spaced in time).

#### *5.2.3.2 Model ability to predict vessel trajectories*

In Figure 5.6, we see that the LRM was able to predict the trajectory of the vessel belonging to MMSI 304927000. When compared with the predicted DKF trajectory in Figure 5.7, we observe that the DKF predictions are more noisy compared to the LRM. The reasons being that matrix $\boldsymbol{K}_t$ has not yet converged (reaching a form of stability) and is over-correcting during the measurement updates. The undersampling rate, $\lambda_{s_i} = 3$, results in longer periods of time where no updates are received by the DKF, and the corresponding matrices taking longer to update and converge.

Due to the constant COG assumption that both the LRM and the DKF makes, we expect linear trajectories to have a constant COG. If the initial COG is incorrect, the speed in the respective directions will be derived incorrectly. The role that the COG plays in the calculations for the LRM is shown in Equation 4.23 an for the DKF in Equation 4.2.

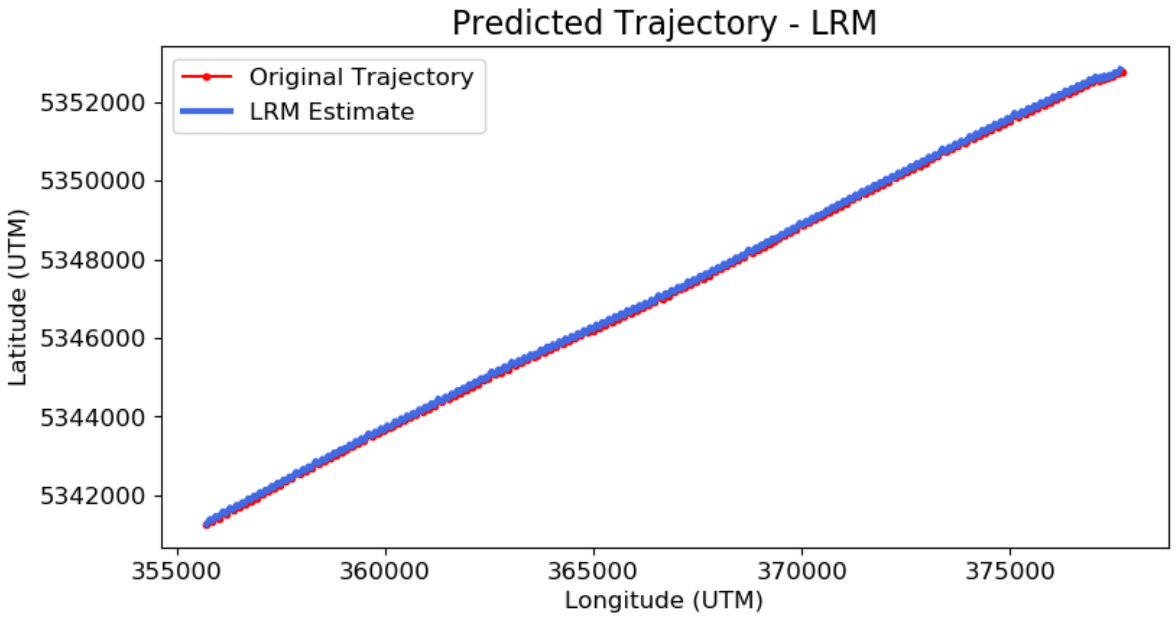


Figure 5.6: LRM trajectory prediction of MMSI 304927000, $\lambda_{s_i} = 3$.

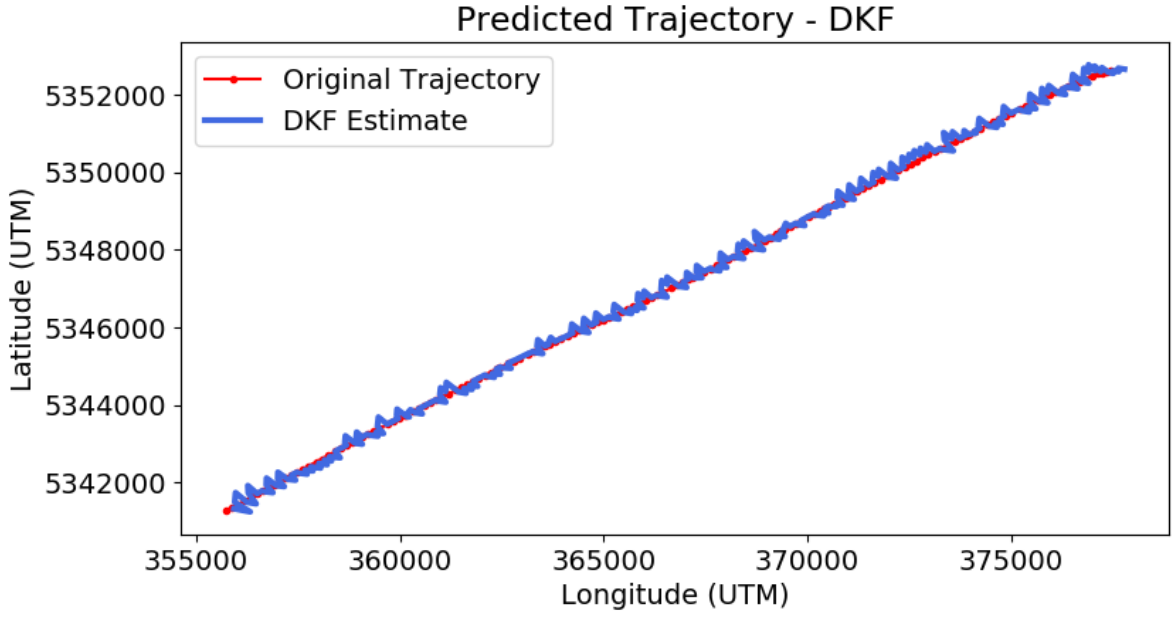


Figure 5.7: DKF trajectory prediction of MMSI 304927000, $\lambda_{s_i} = 3$.

#### *5.2.3.3 Model error comparison*

We now compare the errors made by both models. The errors are calculated by computing the difference in the predicted longitude and observed as well as the predicted latitude and observed.

We also include the Euclidean distance error between the predicted and observed coordinates. We do not use the Haversine distance as our error measure, as the UTM coordinate system is a flat two dimensional projection. The error introduced by using the Euclidean distance will be marginal.

Figure 5.8 and 5.9 denote the distribution of errors for the LRM and the DKF respectively.

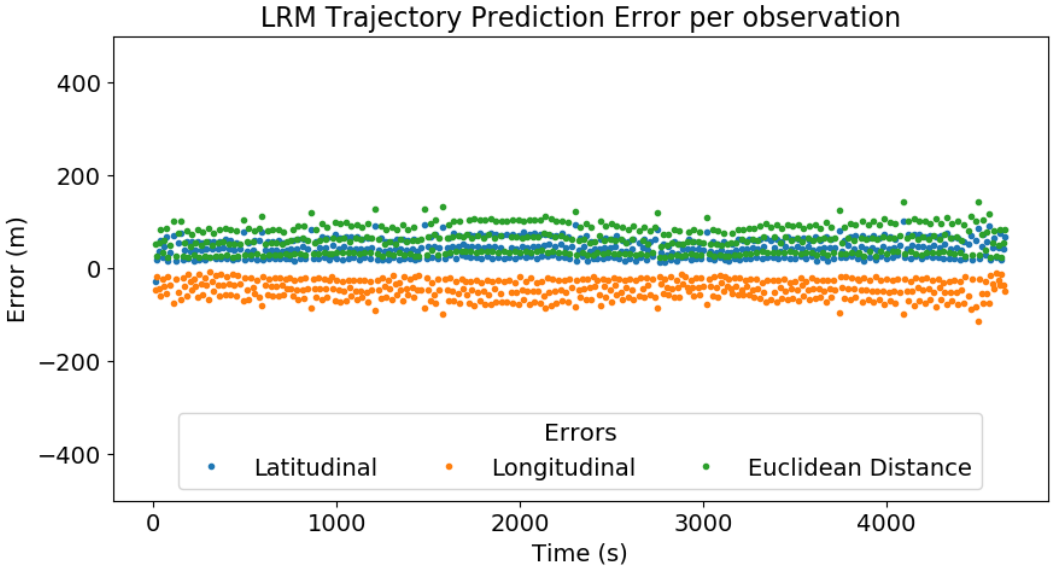


Figure 5.8: LRM error made per observation, when $\lambda_{s_i} = 3$.

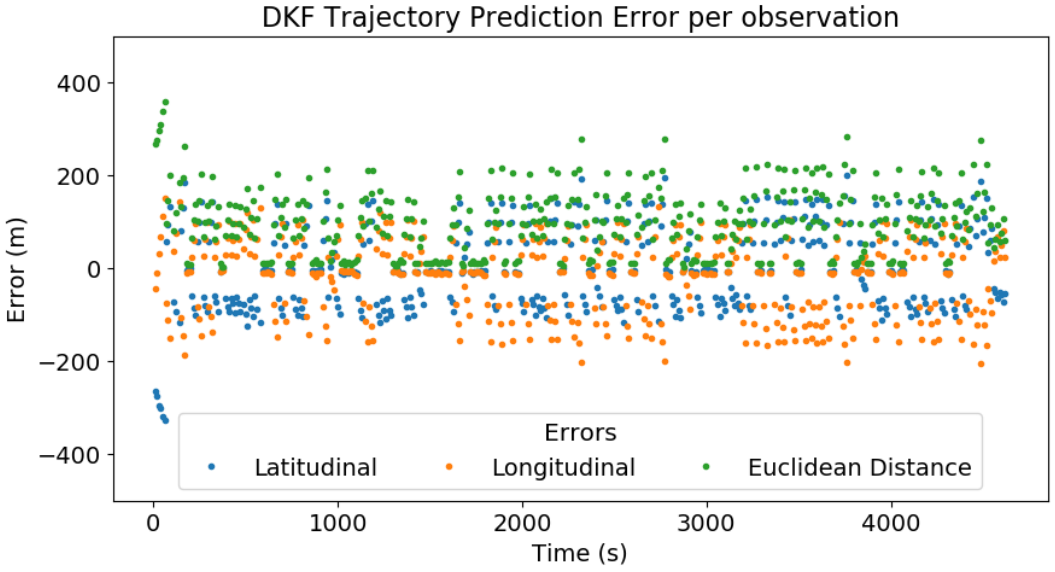


Figure 5.9: DKF error made per observation, when $\lambda_{s_i} = 3$.

We observe that the error distribution of the DKF has a larger variance (more spread out) compared to the LRM. By looking at the spread of the errors over time and how they are centred, we can infer that the errors are probably normally distributed. The error plots confirm that the LRM is outperforming the DKF for MMSI 304927000 when $\lambda_{s_i} = 3$.

Figure 5.10 denotes the overall error comparison (MED) for vessel MMSI 304927000, where the error for each undersampling rate is shown. The error associated with each $\lambda_{s_i}$ is the MED (shown in Equation 5.3).

When comparing the DKF and the LRM for each undersampling rate, we observe that the LRM errors increase less dramatically compared to the errors associated with the DKF. For undersample rates, $7 - 12$, the LRM errors remain nearly constant, whereas the DKF errors keep on increasing.

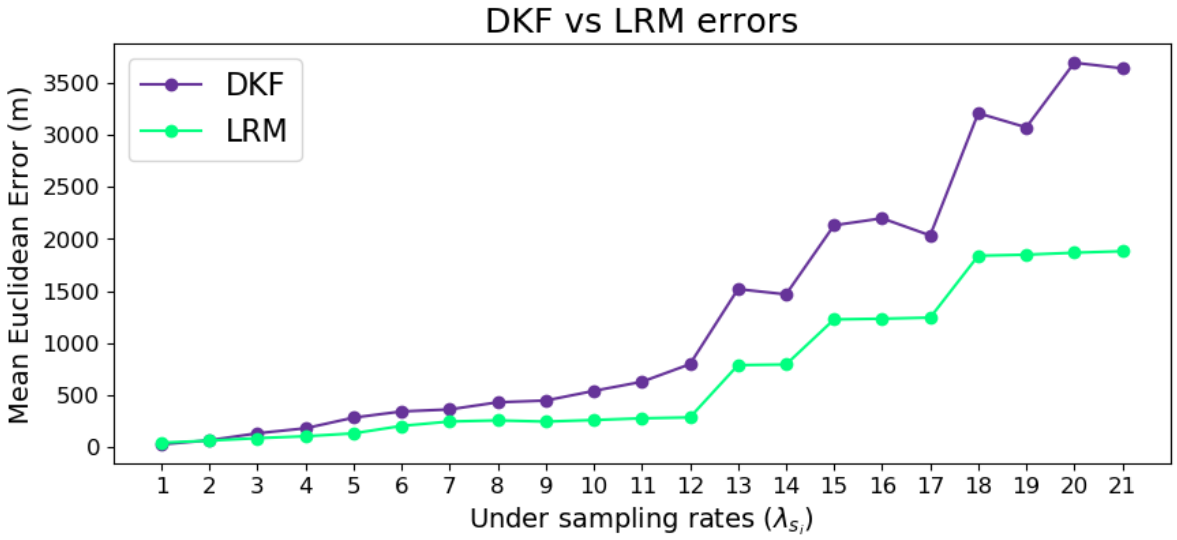


Figure 5.10: Error for each prediction interval $\lambda_{s_i}$ of MMSI 304927000

It will not always be the case that the LRM outperforms the DKF. However, this case study shows an example where it does. Overall our results show that the DKF usually marginally outperforms the LRM method. LRM only outperformed the DKF for 30% of the trajectories in Table C.1, i.e. for only 12 vessels.

## 5.3 *A PRIORI* METHODS

In this section the LRMAC (introduced in Section 4.5) which is an extension of the LRM (introduced in 4.3) is compared to the SPNS (introduced in Section 4.4). The SPNS has a similar programmatic complexity to that of the LRMAC.

As in the case of Section 5.3.1 we first present the experimental setup we employed. This is followed by a comparative study in which we compare the LRMAC and the SPNS with one another (see Section 5.3.3). We then present a specific test case (see Section 5.3.4).

### 5.3.1 Experimental Design

Our experimental design section has a similar outline to the one given in Section 5.2.1. In particular, we will compare the LRM and the LRMAC. This is followed by a comparison study that involves the LRMAC and the SPNS. The following comparison metrics is considered: prediction accuracy and execution time.

#### *5.3.1.1 System Specifications*

The SPNS and LRMAC were tested independently on the same system to avoid any processing or querying bottlenecks. Table 5.1 denotes the system specifications* of the system they were tested on.

| Description | Name | Specification |
|---|---|---|
| CPU | Intel i7-10700K | 8 core 16 threads @ 5.1 GHz |
| Memory | Corsair Vengeance LPX | DDR4 @ 3600 MHz |
| Storage | HP EX920 | NVMe M.2 SSD @ 512 GB |
| Software | Python | 3.8.5 |
| Database | PostgreSQL | 12.6 |
| Database plugin | PostGIS | 3.1.1 |
| Operating System | Ubuntu Desktop | 20.04 LTS |

Table 5.1: Testing system specifications

*The system specifications are not shown for the DKF and the LRM as execution times were not compared. The purpose of the DKF and LRM comparison was to see if the LRM will be a viable solution for trajectory prediction, as stated in our problem statement.

#### *5.3.1.2 Trajectory subsampling method*

A trajectory subsampling method was implemented to ensure that we have a large number of trajectories on which we could test the LRMAC and the SPNS. This method (presented below) effectively creates multiple trajectories from each observed trajectory.

A given trajectory $\boldsymbol{T}_i$ will be subsampled into different time subsets of time lengths $h$. Each time subset's starting observation will differ by one hour compared to the subset that directly precedes it. We refer to this hour difference as a stride step. The symbol $s$ defines the starting point of every stride of $\boldsymbol{T}_i$. Stride values are measured in hours. This method allows us to extract multiple sub trajectories $T_{s,h}$ from $\boldsymbol{T}_i$. Let the number of sub trajectories that can be created with a prediction length of $h$ from $\boldsymbol{T}_i$ be denoted by,

$$\#T_h = \lfloor \max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{\text{hour}} - h + 1 \tag{5.5}$$

and let the total number of sub trajectories from $\boldsymbol{T}_i$ with different starting positions $s$ be denoted by,

$$\#T_{s,h} = \sum_{h=1}^{\lfloor \max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{\text{hour}}} \#T_h \tag{5.6}$$

where,

- $\lfloor \max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{\text{hour}}$ denotes the closest floored hour to the maximum observed time in $\boldsymbol{T}_i$, where $\boldsymbol{t}_{\boldsymbol{T}_i}$ is a vector of all the time steps in $\boldsymbol{T}_i$.
- $h$ refers to the prediction length, where $h \in \{1, 2, ..., \lfloor \max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{\text{hour}}\}$, and
- $s$ denotes to the stride starting position measured in hours, $s \in \{1, 2, ..., \lfloor \max(t_{\boldsymbol{T}_i}) \rfloor_{\text{hour}} - h\}$

This method of sub-trajectory sampling allows us to have multiple trajectories to compare the prediction and time performance between the LRMAC and the SPNS model. The prediction performance is determined by the Haversine distance between the expected spatial location and predicted spatial location at time $t$. The Haversine distance is defined in Equation 3.1, which is the distance on a sphere (in this case, the earth).

In Figure 5.11, the subsampling method is visualised for a six-hour trajectory. Let $s$ denote the starting hour, starting at hour 0, and $h$ represent the prediction length in terms of time. We see that the subsampling method will generate six sub-trajectories each of a length of one hour. The maximum length (in terms of time) for the a trajectory in this example will be six hours.

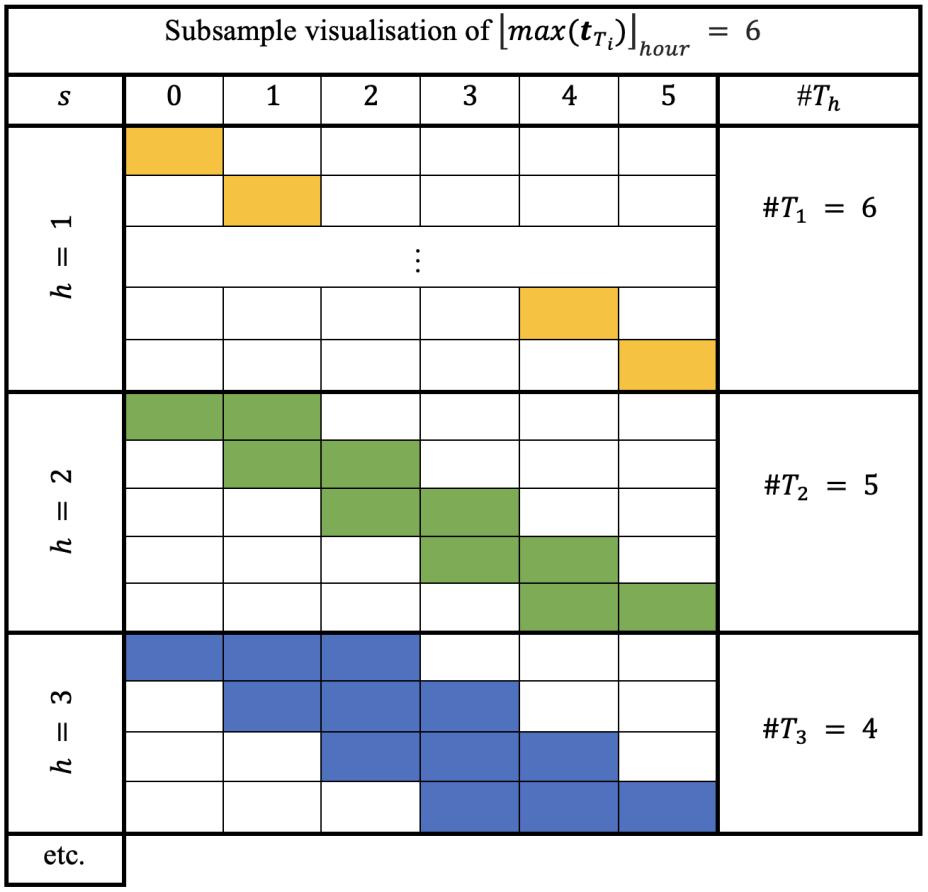


Figure 5.11: Subsampling visualised of a hypothetical six-hour long trajectory

#### *5.3.1.3 SPNS hyperparameters*

The values of the hyperparameters for the SPNS are presented in Table 4.1 Section 4.4.3.

#### *5.3.1.4 LRMAC hyperparameters estimation*

The hyperparameters for the LRMAC were introduced in Section 4.5.2. They are $\eta$ which is the neighbourhood size of the SMs, and $\omega$ which denotes the window size (number of historic

observations to fit the LRMAC on).

A grid search was performed to find the best possible choice for $\omega$ and $\eta$. The combination of $\eta$ and $\omega$ that resulted in the lowest overall Haversine distance error will be selected as the ideal set of hyperparameters. The considered parameter values were: $\eta = [1, 2, 3, 4, 5]$ and $\omega = [1, 3, 5, 7, 9]$, resulting in 20 parameter combination pairs.

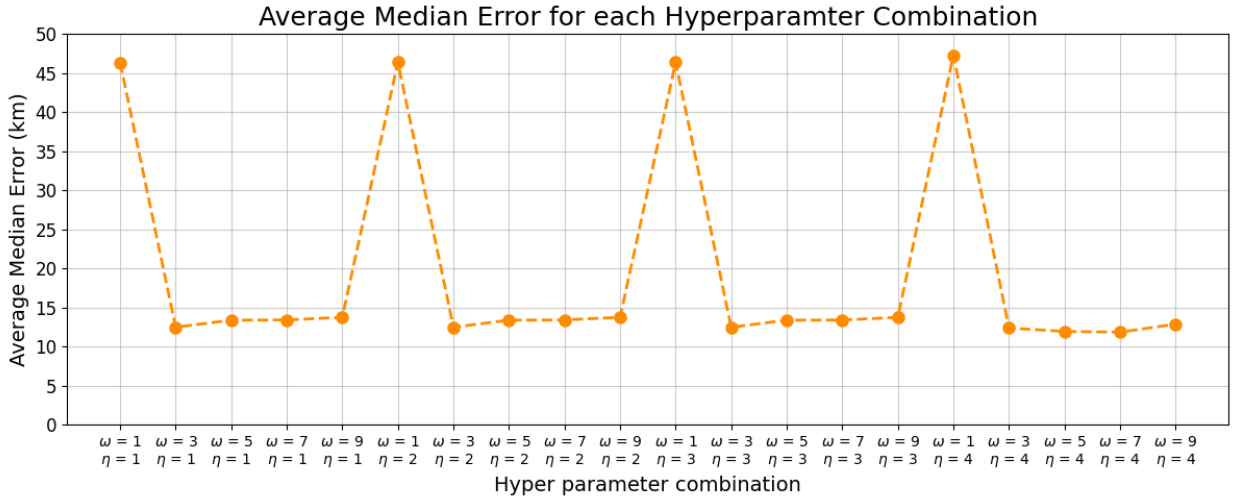


Figure 5.12: Average median Haversine distance error (over the prediction horizons of $5 - 360$ min) for each hyperparameter combination of the LRMAC.

Figure 5.12 depicts the average of the median Haversine distance for all the trajectories for each hyperparameter pair considered. Note that the hyperparameter tuning was performed by utilising all of the trajectories (as in Table C.2) and their sub-trajectories (discussed in Section 5.3.1.2)

The pairs that yielded the best results were $\omega = 3$ and $\eta = 1$, i.e. no other pair resulted in a significant increase in performance. However, it should be noted that although the average Haversine error did decrease when $\eta = 4$, none of the hyperparameter pairs where $\eta$ is equal to 4 was selected. The reason being, even though the error is lower, it is only marginally so, while the computational cost on the other hand would increase (the search neighbourhood grid increases from a $3\times3$ grid to a $9\times9$ grid). The aforementioned hyperparameter choice has the lowest computational complexity of all the other hyperparameter pairs tested. In Section 5.4 the algorithmic complexities are discussed.

Smaller values of $\omega$ mean that our function is more flexible to incorporate new information, as

fewer historic elements are considered for the LRM fit. Large values of $\omega$ will mean that our algorithm is less flexible, using more observed observations from the current trajectory to make a prediction (increase in elements to fit the underlying LRM). Larger values of $\eta$ mean that a larger search neighbourhood has to be considered (an increase in the number of SMs cells that must be included).

#### *5.3.1.5 SPNS Prediction adjustment*

Since the SPNS predicts in constant distance intervals $\Delta l$ instead of constant time intervals as the LRMAC does, we had to adjust the final prediction of the SPNS to allow for an exact comparison at $h$ (prediction horizon) when measuring the performance between the SPNS and the LRMAC.

In order to get the predicted spatial location after $h$ an extra step was added to the SPNS to allow it to predict the spatial location at $h$. The steps are denoted below.

- Let the SPNS predict until the first observation where $\hat{t}^{k+1} > h$. If it is equal to $h$, no further steps are required.
- Calculate the total time that passed (in seconds) between the two predicted observations** $\hat{\boldsymbol{X}}_i^{(k)}$ and $\hat{\boldsymbol{X}}_i^{(k+1)-}$, where the time is indicated by $\hat{t}^k$ and $\hat{t}^{k+1}$ respectively. Note that the two time components should have the following characteristics: $\hat{t}^k < h < \hat{t}^{k+1}$. Let the total time which have passed (in seconds) be denoted by $\Delta t_{|\hat{t}^{k+1} - \hat{t}^k|}$.
- Since the SPNS predict in constant distance intervals and we know the distance between observations $\hat{\boldsymbol{X}}_i^{(k)}$ and $\hat{\boldsymbol{X}}_i^{(k+1)-}$ is denoted by $\Delta l$, we can calculate the distance travelled per second as the time passed during $\Delta l$ between the two observations is now known.
- We also need to calculate the time difference (in seconds) between between $h$ and $\hat{t}^k$, and let it be denoted by $\Delta t_{|h - \hat{t}^k|}$.
- We can now calculate the distance that should be travelled from the observation at $\hat{t}^k$ to the observation at $h$ as follows:

$$\Delta l_{|h-\hat{t}^k|} = \Delta l \times \frac{\Delta t_{|h-\hat{t}^k|}}{\Delta t_{|\hat{t}^{k+1}-\hat{t}^k|}} \tag{5.7}$$

- Given the new interval distance $\Delta l_{|h-\hat{t}^k|}$, we can now re-run the SPNS, where $\Delta l = \Delta l_{|h-\hat{t}^k|}$

**The symbols used are identical to those used in Section 4.4, where the SPNS is introduced.

starting at observation $\hat{\boldsymbol{X}}_i^{(k)-}$ for one iteration. The result will be the predicted spatial location at $h$, where $\hat{t}^{k+1} = h$ as determined by Equation 4.43.

The reason why we do not do straight line imputation and derive the approximate location, is so we can be fair to the SPNS. We allow it to utilise historic information to predict the spatial location up until $h$.

#### *5.3.1.6 Prediction Accuracy Measurement*

Algorithm 5.3 below shows how we calculated the prediction errors for both the LRMAC and SPNS on each sub-trajectory pair $T_{s,h}$. We used the median error results for each predicted time frame $h$ (prediction horizon) to compare the prediction performance between the two methods.

**Algorithm 5.3** *A priori* methods prediction performance comparison

**Set:**
$\omega = 3$ ▷ LRMAC window size
$\eta = 1$ ▷ LRMAC SM neighbours
$\boldsymbol{\epsilon}_{\text{LRMAC}} = [\boldsymbol{\epsilon}_{\text{LRMAC}_1}, \dots, \boldsymbol{\epsilon}_{\text{LRMAC}_{\max(h)}}]$ ▷ Jagged array† for the LRMAC
$\boldsymbol{\epsilon}_{\text{SPNS}} = [\boldsymbol{\epsilon}_{\text{SPNS}_1}, \dots, \boldsymbol{\epsilon}_{\text{SPNS}_{\max(h)}}]$ ▷ Jagged array for the SPNS

**for** $\boldsymbol{T}_i$ **in** $\boldsymbol{T}$ **do**
  Get $\boldsymbol{T}_{s,h}$ which is a set of sub trajectories from $\boldsymbol{T}_i$
  **for** $h$ **in** $[1, 2, \dots, \lfloor \max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{\text{hour}}]$ **do**
    **for** $s$ **in** $[0, 1, 2, \dots, (\lfloor \max(\boldsymbol{t}_{\boldsymbol{T}_i}) \rfloor_{\text{hour}} - h)]$ **do**
      $T_{\text{init}} = \textit{get\_initial}(T_{s,h})$ ▷ Initial observations for both the LRMAC and SPNS
      $\hat{\lambda}_{\text{LRMAC}}, \hat{\phi}_{\text{LRMAC}} = \text{LRMAC}(T_{\text{init}}, \eta, \omega, h)$
      $\hat{\lambda}_{\text{SPNS}}, \hat{\phi}_{\text{SPNS}} = \text{SPNS}(T_{\text{init}}, h)$
      *# Calculate the respective prediction errors to the corresponding h array*
      $\boldsymbol{\epsilon}_{\text{LRMAC}}$ append $\textit{harvdist}(\boldsymbol{T}_{s,h}, \hat{\lambda}_{\text{LRMAC}}, \hat{\phi}_{\text{LRMAC}})$
      $\boldsymbol{\epsilon}_{\text{SPNS}}$ append $\textit{harvdist}(\boldsymbol{T}_{s,h}, \hat{\lambda}_{\text{SPNS}}, \hat{\phi}_{\text{SPNS}})$
      # Where *harvdist* calculates the Harversine distance according Equation 3.1
    **end for**
  **end for**
**end for**

Algorithm 5.3, results in two jagged arrays, one for the SPNS and one for the LRMAC. Both these jagged arrays will contain the errors for each $h$. To evaluate the overall performance per $h$ the median error per array for each $h$ was calculated.

†A *jagged array* refers to an array that contains a collection of arrays. The arrays contained in the jagged array does not have to have the same length. The number of arrays contained in the jagged array used in this thesis will be equal to the maximum trajectory prediction length in terms of hours ($h$).

### 5.3.2 The LRM and LRMAC comparison

The LRMAC is an extension of the LRM, allowing for the prediction of non-linear trajectories. In this section, we do an overall comparison between the LRM and the LRMAC. The Haversine distance is used as our error measure between the expected and predicted location.

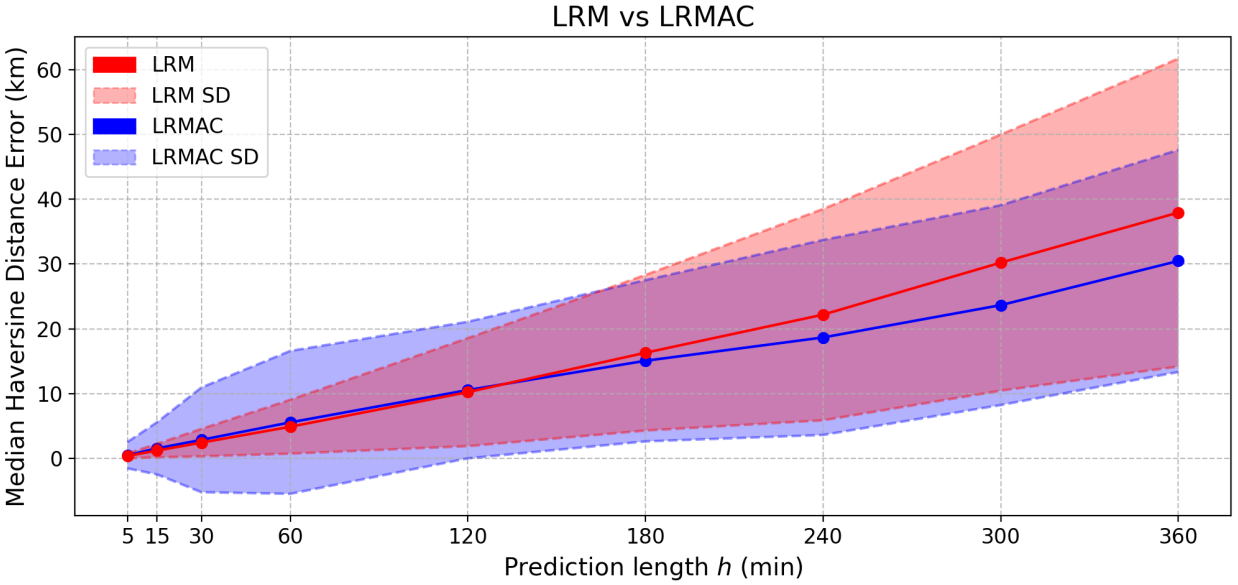


Figure 5.13: LRM and LRMAC error comparison, over a six-hour prediction horizon.

In Figure 5.13, the median Haversine error is shown for both the LRM (red) and LRMAC (blue). The error was calculated over all the sub-trajectories for each time horizon ($h$), together with the standard deviation (SD[‡]) from the median.

When comparing the two methods, we see that the LRMAC has a reduced SD and median error compared to the LRM, making it an improvement over the LRM.

We see that the LRM and LRMAC are not significantly different in terms of short-term prediction accuracy. Shorter prediction periods mean that the subsets of a trajectory will be near-linear as Cargo and Tanker vessels have a slow rate of turn. We also do not expect a vessel trajectory to stay linear for long periods of time, as there are obstacles like landmasses and other vessels.

With respect to longer prediction time horizons ($> 120$ min), the LRMAC has an improved error

[‡]Note that the SD is plotted around the median, instead of the mean absolute deviation. This was done to reflect the algorithmic stability.

of up to 9km. To further support the improvement of the LRMAC over the LRM, when looking at Figure 5.14, we can see that the LRMAC reproduced the trajectory, and the LRM went off course. Figure 5.14 showcases that the model errors can have significantly different meanings, i.e. a 3.48 km difference in predicted error (as seen in Figure 5.14), with one method on the trajectory and the other off-course. We remind the reader that the LRMAC and the LRM both assume a constant velocity model. The background of Figure 5.14 depicts the vessel counts SM $\boldsymbol{K}$, shown in Figure 3.9.

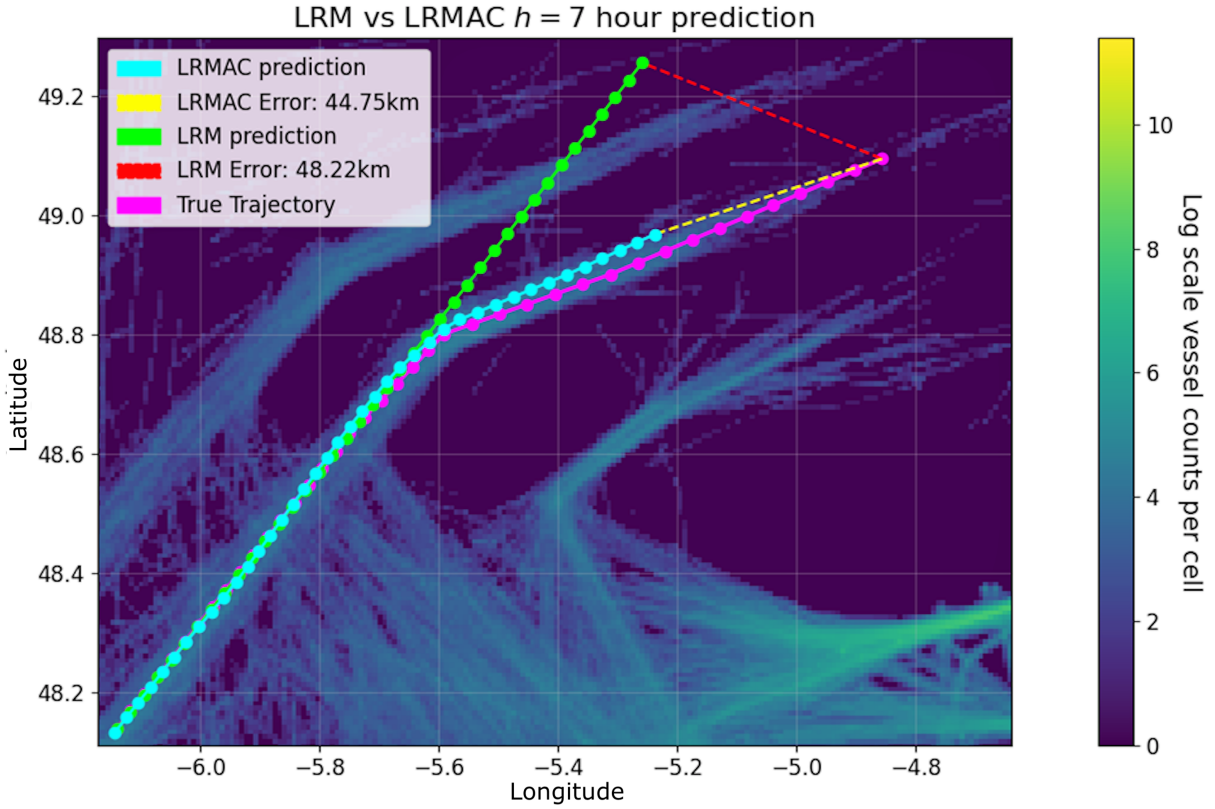


Figure 5.14: LRM vs LRMAC trajectory prediction visualisation projected on an SDM (MMSI 419689000)

We expect that the LRMAC will be able to predict accurately with any SM that has highways exhibiting directionality, as shown in Figure 5.14. Note that the LRMAC algorithm can be applied to linear and non-linear trajectories.

### 5.3.3 The LRMAC and SPNS comparison

In this subsection we compare the LRMAC with the SPNS, as mentioned both are easy to implement and similar in nature.

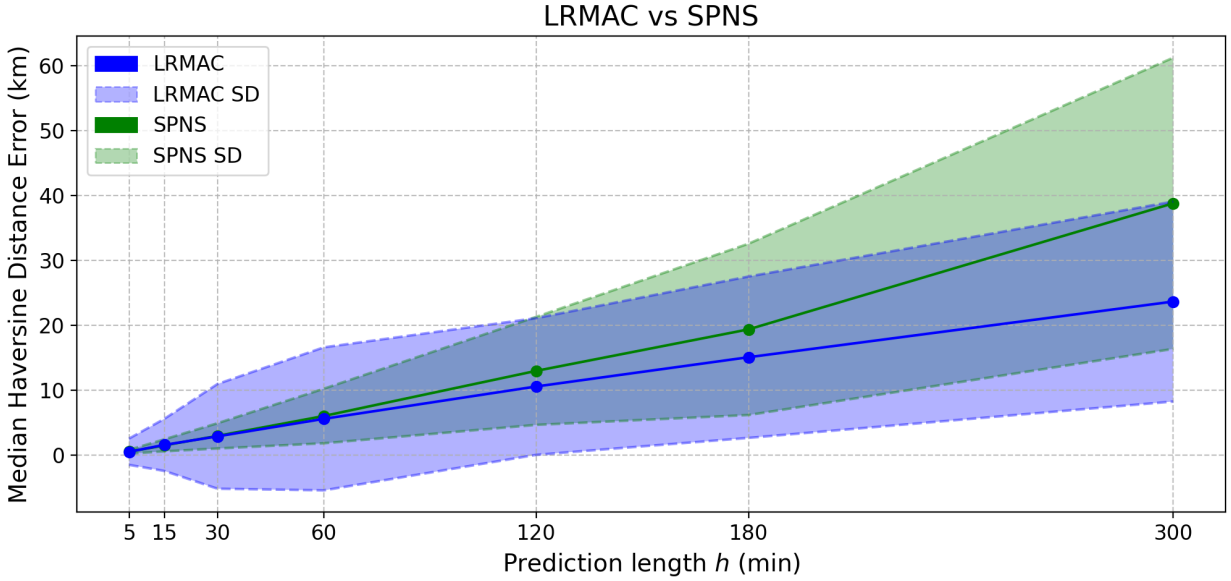


Figure 5.15: SPNS vs LRMAC prediction results

In Figure 5.15, the median prediction error of the LRMAC (blue) is compared to the SPNS (green). Similar to the LRM and LRMAC comparison, the median error and SD for each time horizon ($h$) were calculated. Figure 5.15 is a more accurate and representative comparison of the LRMAC with another method (the SPNS), as both methods follow the historic route, through *a priori* information. This is in contrast with the LRM that went off course, which utilises no *a priori* information.

Looking at Figure 5.15, for short time horizons there are no significant difference in the prediction performance between the LRMAC and the SPNS. However, when considering longer time horizons ($>$ 120min), the median prediction error and SD of the LRMAC is far lower than that of the SPNS. The LRMAC had a reduced error of 17 km for the five-hour prediction time horizon, which in perspective, is a lot. Furthermore, when we look at the SPNS, we see that its SD starts to increase dramatically, compared to the LRMAC's SD, which increases at a lower rate. When looking at the prediction interval [5, 15] min (the recommended time prediction horizon for the SPNS), we see that it has a smaller SD compared to the LRMAC.

| $h$ | LRMAC time | | SPNS time | | Time difference |
|---|---|---|---|---|---|
| **min** | **s** | **min** | **s** | **min** | **min** |
| 5 | 0.11 | 0.0018 | 30.92 | 0.5154 | -0.5135 |
| 15 | 0.33 | 0.0055 | 95.68 | 1.2614 | -1.5892 |
| 30 | 0.66 | 0.0110 | 193.07 | 3.2178 | -3.2068 |
| 60 | 1.31 | 0.0218 | 371.87 | 6.1978 | -6.1760 |
| 120 | 2.64 | 0.0440 | 712.29 | 11.8715 | -11.8275 |
| 180 | 3.98 | 0.0663 | 1026.38 | 17.1063 | -17.0399 |
| 300 | 6.56 | 0.1093 | 1724.46 | 28.7410 | -28.6317 |

Table 5.2: The run time of the LRMAC compared to the SPNS. The median prediction time for each prediction period $h$ is shown.

In Table 5.2, the run time complexity of both methods for each prediction length $h$ is listed. The time was measured over all the sub-trajectories, and the median time is shown. The SPNS takes significantly longer to calculate the predicted trajectory compared to the LRMAC. The speed of the SPNS is limited by the time it takes to query the sets of CNs from the database. Larger database table sizes will lead to more significant query times as there are more observations to search through. The LRMAC uses SMs whose sizes stay fixed, even if more observations are recorded the same LAT and LON range. The SM sizes will only increase if its resolution increase, or the LAT and LAT increase with the cell sizes of the SMs staying fixed.

Looking at Figure 5.15 and Table 5.2, we see that the LRMAC outperforms the SPNS in not only the prediction accuracy for longer time horizons, but also having a significantly shorter execution time. We see that the SPNS has a smaller SD for smaller time horizons compared to the LRMAC.

We believe that in higher density areas closer to harbours, the SPNS will have more accurate prediction results as the set of CNs will only contain vessels moving in the same direction, where the LRMAC would default to the LRM as the *a priori* COG SD value in the corresponding spatial location, will be large. Vessels tend to move slower in areas closer to harbours/ports, and prediction time frames are usually shorter with increased AIS coverage. We think that the SPNS can be used together with the LRMAC. If the LRMAC encounters a cell with a high SD, the SPNS can be deployed until a cell with a lower SD is encountered. A hybrid approach between the SPNS and LRMAC may improve prediction times and prediction accuracy in areas with a significant amount of traffic in different directions. The implementation of such a hybrid approach is deemed out of the scope of the current work.

#### *5.3.3.1 LRMAC limitations*

It should be noted that the LRMAC is limited by the spatial resolution of the SMs it employs. In the case of sparse historic data, the SMs generated from this data would be inaccurate, resulting in a decrease in the performance of the LRMAC. Also, when the prediction time interval between two consecutive predictions is too large, the LRMAC will skip past important *a priori* information and have inaccurate prediction results. Furthermore, the LRMAC was only tested on Cargo and Tanker vessels in this study, as the movement of other vessel types (such as Fishing vessels) is erratic, and the SMs that will reflect this behaviour. The result will be SMs containing values in their cells with an associated high SD.

#### *5.3.3.2 LRMAC applications*

The LRMAC can be used to predict trajectories of vessels or to impute historic trajectories. The LRMAC would be more accurate as observations are recorded, and the observations in the window size $\omega$ is updated. Currently, the predictions of the LRMAC and SPNS were done on the assumption that only the first $\omega$ observations' information will be used, simulating AIS transponders that are switched off for extended periods of time.

### 5.3.4 Case Study - A Comparison between the LRM, LRMAC and SPNS

In this section, we compare the LRM, LRMAC and SPNS with each other on a specific case study. The purpose is to provide further insight into how these methods work, visualising the predicted trajectories after $h = 360$ min.

We do not show the performance on multiple trajectory subsampling rates, as done with the LRM and DKF case study. The reason being, the LRM and DKF testing was done to see if the LRM will be a viable solution to a specific problem set (short term prediction with sporadic observations being recorded). Here the problem is long term prediction without any observation other than the first few (i.e. non-linear trajectories are considered).

#### *5.3.4.1 Model ability to predict vessel trajectories*

In this section, we compare the trajectory prediction performance of both the LRM, LRMAC and SPNS. The LRM is included as the LRMAC is an extension thereof.

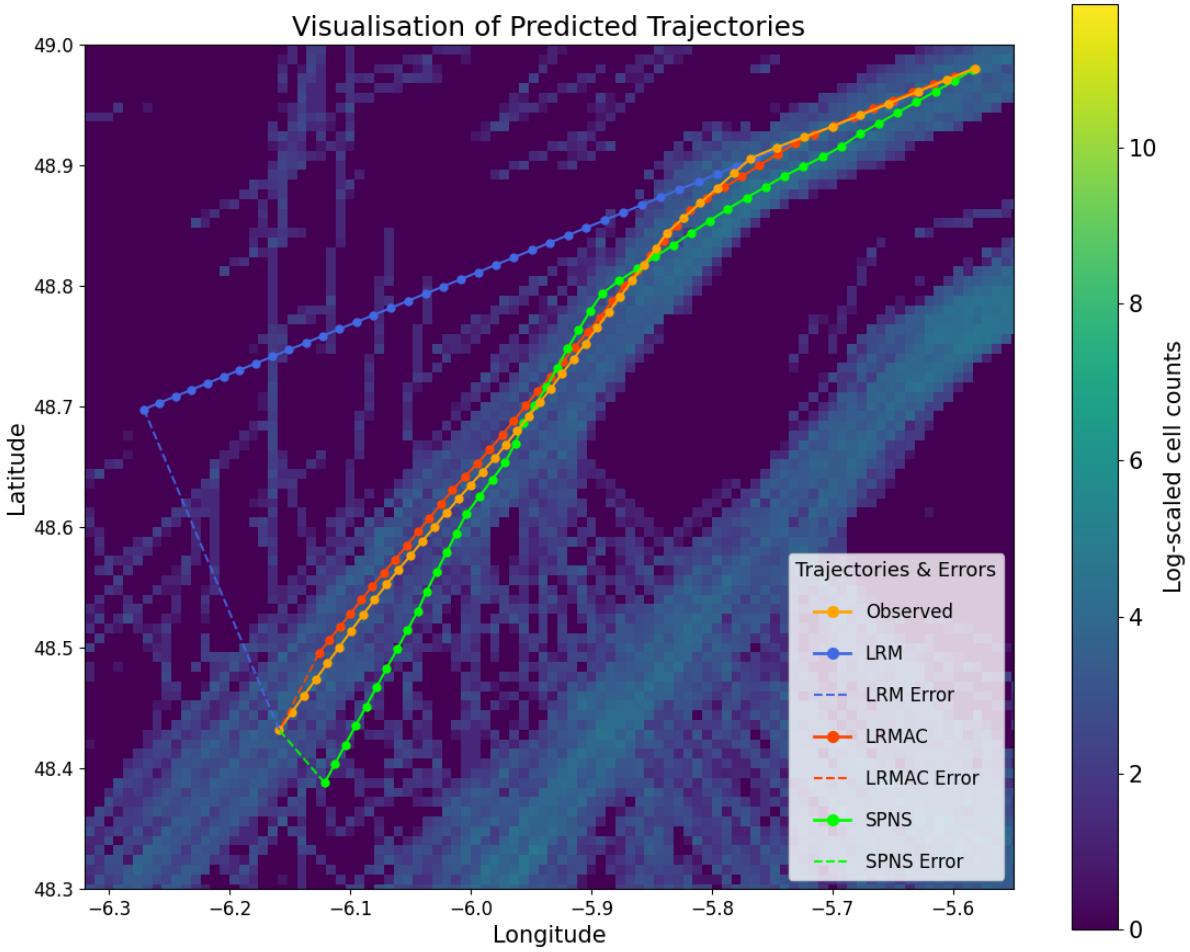


Figure 5.16: Three six-hour predicted trajectories which were predicted by the LRM, LRMAC and SPNS for vessel with MMSI 304805000.

Figure 5.16 denotes the predicted trajectories of the methods being compared. The background of Figure 5.16 represents the vessel counts Spatial Map, visualising the historic information at the respective spatial locations. Note that the SM visualisation is log-scaled. The SM that allows for the non-linear predictions is the COG SM $\mathbf{\Sigma}$, as it can be associated with the historic directions vessels pursued.

Looking at Figure 5.16, we see that both the LRMAC and SPNS prediction remains mostly on-

course. The LRMAC performs the best in terms of recreating the trajectory and slightly under predicts the final location of the vessel. We observe that the SPNS remains on course for most of the prediction. However, it starts to deviate slightly at the end of the six-hour prediction period. As expected, the LRM predicted a linear trajectory and did not follow the expected non-linear course over time.

#### *5.3.4.2 Model error comparison*

The respective prediction errors made by the methods in Figure 5.16 are presented in Table 5.3 below. A zoomed in figure of the trajectory prediction endpoints is also given in Figure 5.17.

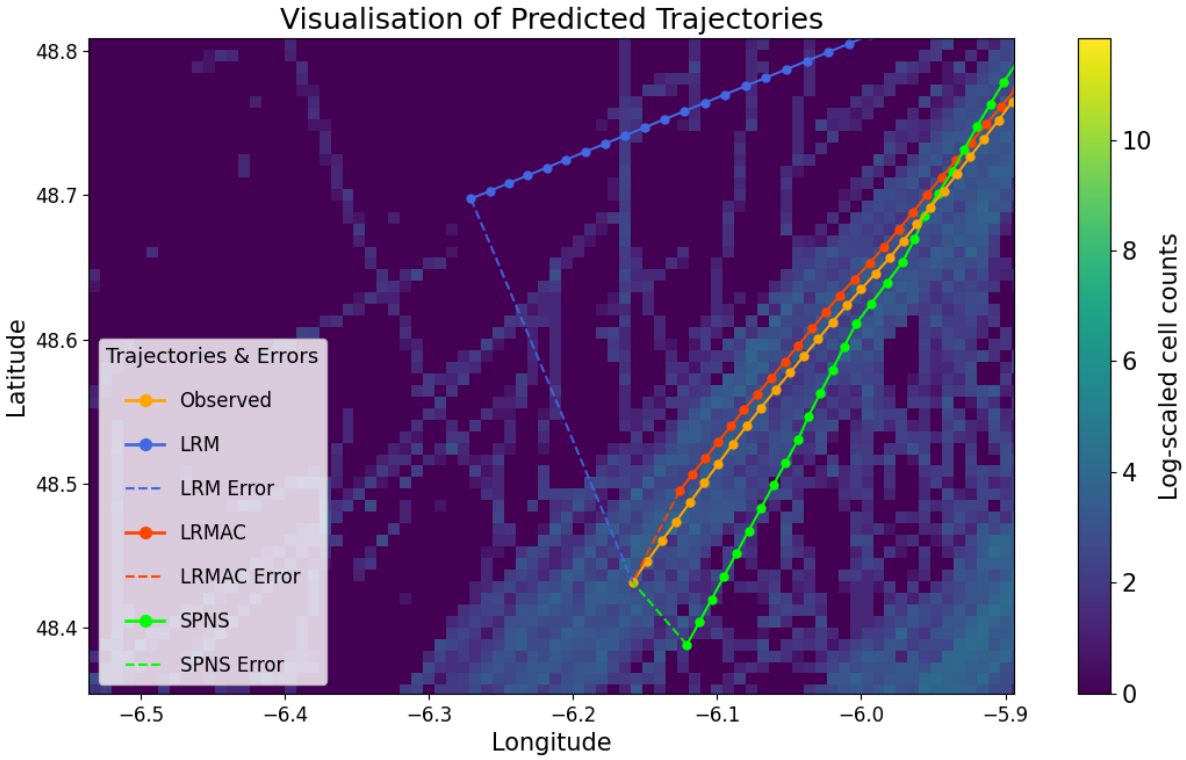


Figure 5.17: Zoomed in view of the LRM, LRMAC and SPNS trajectory predictions of vessel MMSI 304805000.

| Method | Time (h) | Error (km) |
|---|---|---|
| **LRM** | 6 | 32.936 |
| **LRMAC** | 6 | 8.381 |
| **SPNS** | 6 | 6.323 |

Table 5.3: Haversine distance error for each prediction method for vessel MMSI 304805000

Looking at the prediction errors for the six-hour period in Table 5.3, we observe that the SPNS had the lowest error, followed by the LRMAC. The LRM had the largest prediction error as it went off-course. The differences between the LRMAC and the SPNS errors are marginal, especially when the time complexity of both algorithms are taken into consideration (discussed in Section 5.3.3).

We observed that the error does not always indicate the true nature of the predicted trajectories as the LRMAC's predicted trajectory showed a higher degree of similarity to the original trajectory compared to the SPNS. The LRM's limitation of linear trajectories puts it at a disadvantage for long term predictions which is to be expected.

## 5.4 ALGORITHMIC COMPLEXITIES

Algorithmic complexity can be defined as expressing algorithmic running time in terms of the basic computer steps (Dasgupta *et al.*, 2006). It is a more reliable metric than using the time it takes for an algorithm to execute, as algorithms are dependent on the hardware they run on.

The algorithmic complexities are shown below for each of the implemented methods. Algorithmic complexities are expressed using Big-$\mathcal{O}$ notation.

### 5.4.1 DKF

The DKF has a complexity of $\mathcal{O}(n_f^{2.376})$, where $n_f$ refers to the length of the state vector (Montella, 2011). In terms of the DKF, the state vector has a length of $n_f \times 1$, and square matrices of size $n_f \times n_f$. For the DKF implemented in this thesis $n_f = 4$ (see Equation 4.2).

### 5.4.2 LRM

The computational complexity for multiple linear regression[‡] that uses ordinary least squares is $\mathcal{O}(d_l \cdot n_l^2)$ (Lorena *et al.*, 2018). The aforementioned variables are defined as follows:

- $n_l$ is defined as the number of observations used to create the model from.
- $d_l$ is defined as the number of features

[‡] *Multiple linear regression* (MLR), is similar to linear regression, but it has more than one feature that has to be estimated in order to get the best fit. MLR with one feature is simple Linear Regression (the LRM).

The LRM used in this thesis is based on simple linear regression, where the number of features are one ($d_l = 1$), therefore the LRM has a complexity of $\mathcal{O}(n_l^2)$. The number of historic observations used in this thesis was $\omega = 3$ (See section 5.2.1.3). Therefore, the complexity will only increase if the number of historic observations included ($\omega$) increases.

### 5.4.3 LRMAC

The LRMAC is an extension of the LRM, therefore having a complexity of at least $\mathcal{O}(n_m^2)$[§].

We now, however, have to calculate the complexity added to the LRM by the addition of the COG update, which turns it into the LRMAC. Due to the addition of the COG update, the component that contributes the most is the SMs. In order to update the COG, one has to search through the SMs to extract the information at the corresponding index locations.

The index values of the SMs are sorted in ascending order. We remind the reader that $n_{d_r}$ and $n_{d_c}$ indicate the number of rows and columns, respectively. The SMs will always have one of the following properties (dependent on how the SMs were set up):

- $n_{d_r} < n_{d_c}$
- $n_{d_r} = n_{d_c}$
- $n_{d_r} > n_{d_c}$

where $[n_{d_r}, n_{d_c}] \in \mathbb{N}$, and the dimensions of the SMs ($\boldsymbol{K}$, $\boldsymbol{\Psi}$, and $\boldsymbol{\Sigma}$ ) will always be equal.

The extraction of the *a priori* information from the SMs can be done with the Binary search algorithm[¶] (BSA). The BSA has an algorithmic complexity of $\mathcal{O}(\log n_d)$ (Dasgupta *et al.*, 2006), where $n_d$ in the case of the LRMAC is defined as the maximum value between the number of rows and columns in the SMs ($n_d = \max\{n_{d_r}, n_{d_c}\}$).

Since the LRMs complexity is dependent on $\omega$ and the complexity of the added SMs dependent on $n_d$, the complexity of the LRMAC is problem-dependent. Two factors will determine the complexity of the LRMAC:

- Do the chosen resolution of the SMs and the complexity associated with the choice outweigh

[§]Note that in the case for the LRMAC $n_l$ is replaced with $n_m$ for the LRM part of the LRMAC.

[¶]The *Binary search algorithm*, is an algorithm that can efficiently search through a sorted array for a specific value, see Dasgupta *et al.* (2006) for more on the BSA.

the complexity of fitting the LRM?

- Does the complexity of fitting the LRM (size of $\omega$) outweigh the cost that is incurred due to having to search for a value in the SM?

Upon the examination of Figure 5.12, we see that the increase in $\omega$ does not lead to a significant increase in prediction performance of the LRMAC. We, therefore, do not expect the LRMAC to be fitted on more than $\omega = 3$ observations. Therefore, it can be assumed that the overall worst-case complexity of the LRMAC is $\mathcal{O}(\log n_d)$, due to the SMs. In this thesis, $n_m = \omega = 3$, and $n_d = n_{d_r} = n_{d_c} = 1250$. Note that we exclude the complexity of generating the SMs as we are discussing run-time complexity, and the SMs are constructed beforehand.

### 5.4.4 SPNS

The complexity for the SPNS is determined by the extraction of the CN set; obtaining the *a priori* data for its equations.

The SPNS has to search through all the observations in the dataset (to build the CN set) and compare two sets of parameters, i.e. linear search[‖]. The two parameters are the spatial location of an observation (LAT and LON) and whether it is within a predefined radius of the current predicted location (see Section 4.4 for more details). Therefore, the SPNS's worst-case complexity is $\mathcal{O}(n_s)$, where $n_s$ denotes the number of observations present in the dataset, given that the observations are also unsorted.

### 5.4.5 On the algorithmic complexities

In Table 5.4, a summary of the worst case algorithmic complexities of all methods presented in this thesis, is given.

However, these complexities obscure important detail and should not just be taken at face value. The complexities are dependent on the values chosen for each of the parameters.

[‖] *Linear search*, is when each value of an array is checked in sequential order from the start of an array to the finish.

[**] The number 1 864 314, denotes the number of cleaned observations present in the dataset (database that the SPNS had to search through whilst predicting), see Table 3.4 in Section 3.1.1

| Method | Complexity | Description | Values in Thesis |
|---|---|---|---|
| **DKF** | $\mathcal{O}(n_f^{2.376})$ | $n_f$ = state vector dimensions | $n_f = 4$ |
| **LRM** | $\mathcal{O}(n_l^2)$ | $n_l$ = # observations used to fit the LRM | $n_l = 3$ |
| **LRMAC** | $\mathcal{O}(\log n_d)$ | $n_d = \max\{n_{d_r}, n_{d_c}\}$ | $n_d = 1250$ |
| **SPNS** | $\mathcal{O}(n_s)$ | $n_s$ = # observations in the dataset | $n_s = 1\ 864\ 314^{**}$ |

Table 5.4: Complexities of the methods presented in this Thesis.

## 5.5 SUMMARY

In this chapter, we compared the non *a priori* methods with each other as well as the *a priori* methods. The experimental design for the comparison of *a priori* and non *a priori* methods were discussed. For the comparison of the non *a priori* methods, an additional trajectory subsampling method was presented and discussed, which simulated vessels from which minimal updates were received, to test the ability of the DKF and LRM to predict and update once new information were received. A case study for both the non *a priori* and *a priori* methods was also discussed. In the next chapter we conclude the thesis, the drawbacks and trade-offs for each of the methods will be discussed, followed by a conclusion.

# CHAPTER 6

# CONCLUSION AND DISCUSSION

In this final chapter, we give a brief overview of the study conducted. Recall, we first compared the DKF with the LRM on linear trajectories, since the main use case of this study was Cargo and Tanker vessels which have piecewise linear trajectories. We concluded that there was no significant difference between the LRM and the DKF for this use case. After this comparison study yielded promising results, we decided to extend the LRM into the LRMAC, which utilises *a priori* information in the form of spatial maps for improved vessel trajectory prediction. The LRMAC allows for the non-linear predictions of vessel trajectories over extended periods. The method predicts the whole trajectory given the assumption that the vessel will maintain a constant speed from the moment of the first prediction to the last. An *a priori* method from literature, with a similar programmatic complexity to that of the LRMAC, the SPNS was then used to compare the LRMAC to. The LRMAC showed improved performance in long-term prediction accuracy and overall execution speed. The SPNS's and the LRMAC's short-term prediction accuracy was not significantly different.

## 6.1 TRADE-OFFS, DRAWBACKS AND ADVANTAGES

In this section, we discuss the trade-offs, drawbacks and advantages associated with each of the models we investigated in this study.

### 6.1.1 DKF

When predicting, the DKF provides the uncertainty that is associated with a prediction or measurement update in the form of covariance matrices. The matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ of the DKF can be problematic, as the performance of the DKF is dependent on the choice thereof. Choosing $\boldsymbol{Q}$ and $\boldsymbol{R}$ is highly problem-dependent, and if fine-tuned, can greatly improve the accuracy of the DKF. These matrices are used in the predictor and measurement update equations of the DKF, respectively. The matrices are also used when calculating the error covariance $\boldsymbol{P}_t^-$ and Kalman gain $\boldsymbol{K}_t$. If $\boldsymbol{Q}$ and $\boldsymbol{R}$ is chosen in such a way that wrongly represents the problem at hand, the matrices $\boldsymbol{P}_t^-$ and $\boldsymbol{K}_t$ will not converge, and would result in incorrect predictions (see Appendix A, Section A.5

for more details on the DKF parameter tuning).

### 6.1.2 LRM

The LRM is an easy to implement and a straightforward model with minimal hyperparameters to optimise. The only parameter to optimise is the window size $\omega$, a scalar. Compared to the DKF, which has matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. In the presence of outlying SOG observations, the LRM predictions will become unstable, as the estimation of the gradient (slope) of the LRM (see Appendix B) is only fitted on the last $\omega = 3$ observations. Therefore, an outlier will significantly affect the LRM as it is very sensitive to new information. We remind the reader that the LRM estimates the SOG of a vessel in both the latitudinal and longitudinal directions.

Given that the SOG in the respective directions are estimated and the amount of time passed is known ($\Delta k_t = 1$s), the vessel's displacement in the respective directions can be calculated, resulting in the predicted coordinate. If it is known that a particular vessel often records rapid/outlying observations, the $\omega$ can be specifically tuned accordingly, as larger values of $\omega$ reduce the model's flexibility (sensitivity). The LRM is also limited by the fact that it can only predict linear trajectories due to the constant COG assumption based on the last COG update from a vessel. Once a new COG value is observed, the prediction direction will update.

### 6.1.3 SPNS

The SPNS is an algorithm which is highly similar in nature to the LRMAC. The execution speed of SPNS is most affected by the time it takes to query the set of CNs from a database. Larger database table sizes will significantly increase query times as there are more observations to search through. In contrast, the LRMAC uses SMs whose sizes stay fixed, even if more observations become available for a specific geographical area.

### 6.1.4 LRMAC

The LRMAC has minimal added complexity when predicting by the inclusion of SMs for the COG update. However, more pre-processing steps are required to make the data suitable for the SMs which means that when the SMs are constructed from this data, they will be a good representation of the *a priori* information in a given spatial area. The LRMAC, however, is also limited by the

spatial resolution of the SMs. In the case of limited historic data, the SMs generated from this data would be inaccurate, resulting in a decrease in performance of the LRMAC; with minimal *a priori* information, the LRMAC will default to the LRM. Additionally, when the prediction time interval between two consecutive predictions is too large, the LRMAC will skip past important *a priori* information resulting in inaccurate prediction results. Due to the fact that the size of the SMs stay the same no matter the number of observations recorded, the LRMAC's run-time stays constant with an increase in data points. In contrast, the SPNS has more observations to search through, slowing its execution time. The size of an SM will only increase if its resolution increases, or the latitude and longitude span increases (the cell size/resolution stays fixed). The SMs size is not affected by the number of historic observations recorded.

## 6.2 CONCLUSION

Recall that the main research questions of this study were presented in Section 1.2. In this section, we investigate whether we have adequately addressed them.

We have shown that the LRM does not perform significantly worse than other more complex prediction models for the Cargo and Tanker use case, thus being a viable option when we have linear trajectories. As new observations are recorded from the target vessel, the LRM will adapt to the incoming data as it is refitted on the latest $\omega$ observations.

There is no significant difference in the performance of the DKF (Section 4.2) and the LRM (Section 4.3) when they were used to predict observations over short prediction intervals. The results from this comparison was presented in Section 5.2.2. Furthermore, the LRM is easier to configure than the DKF, since the LRM only has one parameter $\omega$ to optimise (the window size) compared to the non-trivial optimisation of both the $\mathbf{R}$ and $\mathbf{Q}$ matrices in the case of the DKF. In conclusion, the LRM approach is a sufficient short term trajectory prediction algorithm for the use case of Cargo and Tanker vessels (since the majority of the route segments of these vessels are linear). This then answers research question one.

The LRMAC (see Section 4.5) was created by extending the LRM, by incorporating *a priori* information in the form of SMs (see Section 3.2); allowing for improved vessel trajectory prediction for long periods of time. We found that the LRMAC out performs an existing method that also use *a*

*priori* information, with respect to long term predictions. The LRMAC achieves good prediction accuracies with a relatively low time and algorithmic complexity, especially as the volume of historic data increases. Answering both research questions two and three.

The LRMAC had a smaller incurred prediction error and associated standard deviation than the LRM. The LRMAC could predict a trajectory which were more representative of the actual trajectory, where the LRM went off course. The LRMAC was compared to the SPNS (see Section 5.3.3), which is very similar in nature to the LRMAC, for the use case of predicting Cargo and Tanker vessel trajectories up to six hours into the future. The LRMAC outperformed the SPNS both in terms of the prediction accuracy as well as execution time (see Section 5.3.3 and 5.4). The LRMAC, therefore, can be used to predict trajectories of vessels or impute vessel trajectories. Future work includes exploring the possibility of a hybrid approach between the LRMAC and SPNS.

Lower complexity models still have an important role to play in modern problems. Modern problems do not always need to be solved by the newest, most popular approaches; as is clear from the case study presented here, a simple linear model with some added information still results in a useful and scalable algorithm that can achieve a sufficient prediction accuracy.

We have to ask ourselves, is it really necessary to use the most advanced, cutting edge algorithm currently available for the specific problem at hand? Or should we start simple and work our way up, considering more advanced algorithms? The trade-off between using complex models over less complex models has to be considered, and whether the increase in performance of using a more complex model is significant enough to be justified. From a sustainability perspective, more complex models incur a higher cost of implementation and have a larger environmental impact when compared to less complex models. Therefore, we conclude with a question: "Is it necessary to use a more complex model over a less complex model, with an insignificant increase in performance?"

# REFERENCES

Achiri, L., Guida, R. and Iervolino, P. (2018). SAR and AIS fusion for maritime surveillance. In: *2018 IEEE 4th International Forum on Research and Technology for Society and Industry (RTSI)*, pp. 1–4. IEEE.

Alizadeh, D., Alesheikh, A.A. and Sharif, M. (2021). Vessel trajectory prediction using historical automatic identification system data. *Journal of Navigation*, vol. 74, no. 1, p. 156–174.

Androjna, A., Perkovič, M., Pavic, I. and Mišković, J. (2021). AIS data vulnerability indicated by a spoofing case-study. *Applied Sciences*, vol. 11, no. 11, p. 5015.

Anneken, M., Jousselme, A.-L., Robert, S. and Beyerer, J. (2018). Synthetic trajectory extraction for maritime anomaly detection. In: *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 1048–1053.

Balduzzi, M., Pasta, A. and Wilhoit, K. (2014). A security evaluation of ais automated identification system. In: *Proceedings of the 30th Annual Computer Security Applications Conference*, p. 436–445. Association for Computing Machinery, New York, NY, USA.

Bautista-Sánchez, R., Barbosa-Santillan, L.I. and Sánchez-Escobar, J.J. (2021). Method for select best AIS data in prediction vessel movements and route estimation. *Applied Sciences*, vol. 11, no. 5, p. 2429.

Besse, P.C., Guillouet, B., Loubes, J.-M. and Royer, F. (2016). Review and perspective for distance-based clustering of vehicle trajectories. *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3306–3317.

Brusch, S., Lehner, S., Fritz, T., Soccorsi, M., Soloviev, A. and van Schie, B. (2010). Ship surveillance with TerraSAR-X. *IEEE Transactions on Geoscience and Remote sensing*, vol. 49, no. 3, pp. 1092–1103.

Burger, C.N., Grobler, T.L. and Kleynhans, W. (2020). Discrete kalman filter and linear regression comparison for vessel coordinate prediction. In: *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pp. 269–274.

Burger, J.E. (2021). Artisanal depiction of the AIS communication network. Available at: `https://www.duardburger.myportfolio.com`

Campello, R.J., Moulavi, D. and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In: *Pacific-Asia conference on knowledge discovery and data mining*, pp. 160–172. Springer.

Chan, Y.K. and Koo, V. (2008). An introduction to synthetic aperture radar (SAR). *Progress In Electromagnetics Research B*, vol. 2, pp. 27–60.

Chaturvedi, S.K. (2019). Study of synthetic aperture radar and automatic identification system for ship target detection. *Journal of Ocean Engineering and Science*, vol. 4, no. 2, pp. 173–182.

Chaturvedi, S.K., Yang, C., Ouchi, K. and Shanmugam, P. (2012). Ship recognition by integration of SAR and AIS. *Journal of Navigation*, vol. 65, no. 2, p. 323–337.

Chen, G. (1992). Introduction to random signals and applied Kalman filtering. *International Journal of Adaptive Control and Signal Processing*, vol. 6, no. 5, pp. 516–518.

Chen, L. and Ng, R. (2004). On the marriage of lp-norms and edit distance. In: *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, p. 792–803.

Chen, L., Özsu, M.T. and Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 491–502.

Chen, X., Wang, S., Shi, C., Wu, H., Zhao, J. and Fu, J. (2019). Robust ship tracking via multi-view learning and sparse representation. *Journal of Navigation*, vol. 72, no. 1, p. 176–192.

Curlander, J.C. and McDonough, R.N. (1991). *Synthetic Aperture Radar Systems and Signal Processing*. Wiley, New York.

Cutrona, L. (1990). Synthetic aperture radar. *Radar handbook*, vol. 2, pp. 2333–2346.

Dalsnes, B.R., Hexeberg, S., Flåten, A.L., Eriksen, B.H. and Brekke, E.F. (2018). The neighbor course distribution method with Gaussian Mixture Models for AIS-based vessel trajectory prediction. In: *2018 21st International Conference on Information Fusion (FUSION)*, pp. 580–587.

Dasgupta, S., Papadimitriou, C. and Vazirani, U. (2006). *Algorithms*. McGraw-Hill Higher Education.

Drinkwater, M.R., Kwok, R. and Rignot, E. (1990). Synthetic aperture radar polarimetry of sea ice. In: *10th Annual International Symposium on Geoscience and Remote Sensing*, pp. 1525–1528. IEEE.

Dzvonkovskaya, A. and Rohling, H. (2010). HF radar performance analysis based on AIS ship information. In: *2010 IEEE Radar Conference*, pp. 1239–1244.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, p. 226–231. AAAI Press.

Fang, L., Jiang, Q., Shi, J. and Zhou, B. (2020*a*). TPNet: Trajectory proposal network for motion prediction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6797–6806.

Fang, S., Wang, Y., Gou, B. and Xu, Y. (2020*b*). Toward future green maritime transportation: An overview of seaport microgrids and all-electric ships. *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 207–219.

Fikioris, G., Patroumpas, K. and Artikis, A. (2020*a*). Optimizing vessel trajectory compression. In: *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pp. 281–286.

Fikioris, G., Patroumpas, K., Artikis, A., Paliouras, G. and Pitsikalis, M. (2020*b*). Fine-tuned compressed representations of vessel trajectories. In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, p. 2429–2436. Association for Computing Machinery, New York, NY, USA.

Forti, N., Millefiori, L.M., Braca, P. and Willett, P. (2020). Prediction of vessel trajectories from AIS data via sequence-to-sequence recurrent neural networks. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8936–8940.

Gao, D., Zhu, Y., Zhang, J., He, Y., Yan, K. and Yan, B. (2021). A novel MP-LSTM method for ship trajectory prediction based on AIS data. *Ocean Engineering*, vol. 228, p. 108956.

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep learning*. MIT press.

Gordon, N.J., Salmond, D.J. and Smith, A.F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal processing)*, vol. 140, no. 2, pp. 107–113.

Grobler, T. and Kleynhans, W. (2019). Extracting high-volume traffic routes from AIS spatial distribution maps. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 10031–10034.

Gruber, C., Gruber, T., Krinninger, S. and Sick, B. (2009). Online signature verification with support vector machines based on LCSS kernel functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 4, pp. 1088–1100.

Gudmundsson, J. and Valladares, N. (2014). A GPU approach to subtrajectory clustering using the fréchet distance. *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 924–937.

Har-Peled, S. *et al.* (2002). New similarity measures between polylines with applications to morphing and polygon sweeping. *Discrete & Computational Geometry*, vol. 28, no. 4, pp. 535–569.

Harchowdhury, A., Sarkar, B.K., Bandyopadhyay, K. and Bhattacharya, A. (2012). Generalized mechanism of SOTDMA and probability of reception for satellite-based AIS. In: *2012 5th International Conference on Computers and Devices for Communication (CODEC)*, pp. 1–4.

Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P. *et al.* (2020). Array programming with NumPy. *Nature*, vol. 585, no. 7825, pp. 357–362.

Hart, E. and Timmis, J. (2008). Application areas of AIS: The past, the present and the future. *Applied Soft Computing*, vol. 8, no. 1, pp. 191–201.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer.

Hexeberg, S. (2017). *AIS-based Vessel Trajectory Prediction for ASV Collision Avoidance*. Master's thesis, Norwegian University of Science and Technology.

Hexeberg, S., Flåten, A.L., Eriksen, B.-O.H. and Brekke, E.F. (2017). AIS-based vessel trajectory prediction. In: *2017 20th International Conference on Information Fusion (Fusion)*, pp. 1–8.

Hovland, H.A., Johannessen, J.A. and Digranes, G. (1994). Slick detection in SAR images. In: *Proceedings of IGARSS'94-1994 IEEE International Geoscience and Remote Sensing Symposium*, vol. 4, pp. 2038–2040. IEEE.

International Maritime Organization (2020). Long-range identification and tracking system - technical documentation (part 1). *MSC.1/Circ.1259/Rev.8*, pp. 1–206.

International Telecommunication Union (2012). Assignment and use of identities in the maritime mobile service. *Recommendation ITU-R M.585-6*, pp. 1–10.

Iphar, C., Ray, C. and Napoli, A. (2020). Data integrity assessment for maritime anomaly detection. *Expert Systems with Applications*, vol. 147, p. 113219.

Jacobs, O.L.R. (1974). *Introduction to control theory*. Oxford University Press, USA.

James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). *An introduction to statistical learning*, vol. 112. Springer.

Jaskolski, K. (2017). Automatic identification system (AIS) dynamic data estimation based on discrete kalman filter (KF) algorithm. *Zeszyty Naukowe Akademii Marynarki Wojennej*, vol. 58, no. 4 (211), pp. 71–87.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, vol. 83 Series D, pp. 34–45.

Kanjir, U. (2019). Detecting migrant vessels in the mediterranean sea: Using Sentinel-2 images to aid humanitarian actions. *Acta Astronautica*, vol. 155, pp. 45–50.

Kanjir, U., Greidanus, H. and Oštir, K. (2018). Vessel detection and ification from spaceborne optical images: A literature survey. *Remote Sensing of Environment*, vol. 207, pp. 1–26.

Kim, K. and Lee, K.M. (2018). Deep learning-based caution area traffic prediction with automatic identification system sensor data. *Sensors*, vol. 18, no. 9.

Kleynhans, W., Salmon, B., Schwegmann, C. and Seotlo, M. (2013). Ship detection in south african oceans using a combination of SAR and historic LRIT data. In: *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*, pp. 1521–1524.

Kong, J., Yueh, S., Lim, H., Shin, R. and Van Zyl, J. (1990). Classification of earth terrain using polarimetric synthetic aperture radar images. *Progress In Electromagnetics Research*, vol. 3, pp. 327–370.

Koppe, W., Bach, K. and Lumsdon, P. (2014). Benefits of terra-SAR-X-PAZ constellation for maritime surveillance. In: *EUSAR 2014; 10th European Conference on Synthetic Aperture Radar*, pp. 1–4. VDE.

Kumar, M. (1988). World geodetic system 1984: A modern and accurate global reference frame. *Marine Geodesy*, vol. 12, no. 2, pp. 117–126.

Lang, H., Wu, S. and Xu, Y. (2018). Ship classification in SAR images improved by AIS knowledge transfer. *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 3, pp. 439–443.

Li, D., Liu, H. and Ng, S. (2020). VC-GAN: classifying vessel types by maritime trajectories using generative adversarial networks. In: *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 923–928.

Liu, J., Shi, G. and Zhu, K. (2020). Online multiple outputs least-squares support vector regression model of ship trajectory prediction based on automatic information system data and selection mechanism. *IEEE Access*, vol. 8, pp. 154727–154745.

Lorena, A.C., Maciel, A.I., de Miranda, P.B., Costa, I.G. and Prudêncio, R.B. (2018). Data complexity meta-features for regression problems. *Machine Learning*, vol. 107, no. 1, pp. 209–246.

Lynne, G.J. and Taylor, G.R. (1986). Geological assessment of SIR-B imagery of the Amadeus Basin, N. T., Australia. *IEEE Transactions on Geoscience and Remote Sensing*, vol. GE-24, no. 4, pp. 575–581.

Ma, Y., Yue, B., Chenier, R., Omari, K. and Henschel, M. (2021). Nearshore bathymetry estimation using synthetic aperture radar (SAR) imagery. *Canadian Journal of Remote Sensing*, vol. 47, no. 6, pp. 1–12.

Machado, T.M.C. (2018). *Maritime modular anomaly detection framework*. Ph.D. thesis, University Institute of Lisbon.

Magdy, N., Sakr, M.A., Mostafa, T. and El-Bahnasy, K. (2015). Review on trajectory similarity measures. In: *2015 IEEE seventh international conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 613–619. IEEE.

Maresca, S., Braca, P., Horstmann, J. and Grasso, R. (2014). Maritime surveillance using multiple high-frequency surface-wave radars. *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 5056–5071.

Maybeck, P.S. (1982). *Stochastic models, estimation, and control*. Academic press, New York, NY, USA.

Mazzarella, F., Arguedas, V.F. and Vespe, M. (2015*a*). Knowledge-based vessel position prediction using historical AIS data. In: *2015 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–6.

Mazzarella, F., Vespe, M. and Santamaria, C. (2015*b*). SAR ship detection and self-reporting data fusion based on traffic knowledge. *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 8, pp. 1685–1689.

Meraner, A., Ebel, P., Zhu, X.X. and Schmitt, M. (2020). Cloud removal in Sentinel-2 imagery using a deep residual neural network and SAR-optical data fusion. *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 166, pp. 333–346.

Milios, A., Bereta, K., Chatzikokolakis, K., Zissis, D. and Matwin, S. (2019). Automatic fusion of satellite imagery and AIS data for vessel detection. In: *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–5.

Montella, C. (2011). The Kalman Filter and related algorithms: A literature review. pp. 1–17.

Murray, B. and Perera, L.P. (2020). A dual linear autoencoder approach for vessel trajectory prediction using historical AIS data. *Ocean Engineering*, vol. 209, p. 107478.

Murray, B. and Perera, L.P. (2021). An AIS-based deep learning framework for regional ship behaviour prediction. *Reliability Engineering & System Safety*, vol. 215, p. 107819.

Müller, M. (2007). *Dynamic Time Warping*, pp. 69–84. Springer Berlin Heidelberg.

Natale, F., Gibin, M., Alessandrini, A., Vespe, M. and Paulrud, A. (2015). Mapping fishing effort through AIS data. *PLOS ONE*, vol. 10, no. 6, pp. 1–16.

Notteboom, T., Pallis, A. and Rodrigue, J. (2020). *Port Economics, Management and Policy*. Routledge.

Octavian, A. and Jatmiko, W. (2020). Designing intelligent coastal surveillance based on big maritime data. In: *2020 International Workshop on Big Data and Information Security (IWBIS)*, pp. 1–8.

Pallotta, G., Horn, S., Braca, P. and Bryan, K. (2014). Context-enhanced vessel prediction based on Ornstein-Uhlenbeck processes using historical AIS traffic patterns: Real-world experimental results. In: *17th International Conference on Information Fusion (FUSION)*, pp. 1–7.

Pallotta, G., Vespe, M. and Bryan, K. (2013). Traffic route extraction and anomaly detection from AIS data. In: *International COST MOVE Workshop on Moving Objects at Sea, Brest, France*, pp. 1–4.

Papi, F., Podt, M., Boers, Y., Battistello, G. and Ulmke, M. (2012). On constraints exploitation for particle filtering based target tracking. In: *2012 15th International Conference on Information Fusion*, pp. 455–462. IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. *et al.* (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, vol. 12, pp. 2825–2830.

Perera, L.P., Oliveira, P. and Guedes Soares, C. (2012). Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction. *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1188–1200.

Pitsikalis, M., Do, T.-T., Lisitsa, A. and Luo, S. (2021). Logic rules meet deep learning: A novel approach for ship type classification.

PostGIS (2021). PostGIS.
Available at: `http://postgis.net/`

PostgreSQL (2021). PostgreSQL:the worlds most advanced open source database.
Available at: `http://postgresql.org/`

QGIS Development Team (2021). *QGIS Geographic Information System*. QGIS Association.
Available at: `https://www.qgis.org`

Ray, C., Dréo, R., Camossi, E., Jousselme, A.-L. and Iphar, C. (2019). Heterogeneous integrated dataset for maritime intelligence, surveillance, and reconnaissance. *Data in Brief*, vol. 25, p. 104141.

Rice, J.A. (2006). *Mathematical statistics and data analysis*. Cengage Learning.

Ristic, B., La Scala, B., Morelande, M. and Gordon, N. (2008). Statistical analysis of motion patterns in AIS data: Anomaly detection and motion prediction. In: *2008 11th International Conference on Information Fusion*, pp. 1–7. IEEE.

Rodrigue, J., Comtois, C. and Slack, B. (2016). *The geography of transport systems*. Routledge.

Rong, H., Teixeira, A. and Guedes Soares, C. (2019). Ship trajectory uncertainty prediction based on a Gaussian process model. *Ocean Engineering*, vol. 182, pp. 499–511.

Rong, H., Teixeira, A. and Guedes Soares, C. (2020). Data mining approach to shipping route characterization and anomaly detection based on ais data. *Ocean Engineering*, vol. 198, p. 106936.

Russell, S. and Norvig, P. (2002). *Artificial intelligence: a modern approach*. Prentice Hall.

Saputra, H., Sototo, S.W., MuftiFathonah, M., Istardi, D., Atmaja, A.B.K. *et al.* (2018). Development of automatic identification system AIS for vessels traffic monitoring in the strait of Singapore and Batam waterways. *Journal of Ocean, Mechanical and Aerospace -science and engineering*, vol. 51, no. 1, pp. 7–13.

Saunders, C., Gammerman, A. and Vovk, V. (1998). Ridge regression learning algorithm in dual variables. In: *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, p. 515–521. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Schwegmann, C. *et al.* (2018). *Advanced ship detection methods in Synthetic Aperture Radar imagery.* Ph.D. thesis, University of Pretoria.

Schwegmann, C.P., Kleynhans, W. and Salmon, B.P. (2017). Synthetic aperture radar ship detection using haar-like features. *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 2, pp. 154–158.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In: *3rd International Conference on Learning Representations*, pp. 1–14.

Storvold, R., Malnes, E., Larsen, Y., Høgda, K., Hamran, S., Mueller, K. and Langley, K. (2006). SAR remote sensing of snow parameters in norwegian areas - current status and future perspective. *Journal of Electromagnetic Waves and Applications*, vol. 20, no. 13, pp. 1751–1759.

Suo, Y., Chen, W., Claramunt, C. and Yang, S. (2020). A ship trajectory prediction framework based on a recurrent neural network. *Sensors*, vol. 20, no. 18.

Suykens, J.A., Van Gestel, T., De Brabanter, J., De Moor, B. and Vandewalle, J.P. (2002). *Least squares support vector machines.* World scientific.

Suykens, J.A. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, vol. 9, no. 3, pp. 293–300.

Tampakis, P. (2020). Big mobility data analytics: Algorithms and techniques for efficient trajectory clustering. In: *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pp. 244–245.

Tang, H., Yin, Y. and Shen, H. (2019). A model for vessel trajectory prediction based on long short-term memory neural network. *Journal of Marine Engineering & Technology*, pp. 1–10.

Theodoropoulos, G.S., Tritsarolis, A. and Theodoridis, Y. (2019). EvolvingClusters: Online discovery of group patterns in enriched maritime data. In: *International Workshop on Multiple-Aspect Analysis of Semantic Trajectories*, pp. 50–65. Springer.

Ulaby, F.T., Moore, R.K. and Fung, A.K. (1981). *Microwave remote sensing: Active and passive*. Artech House Publishers.

United Nations (1958). *Treaty Collection - Chapter XII, Navigation*, vol. 289 & 1520.

United Nations (1980). International convention for the safety of life at sea. *Treaty Series*, vol. 1184, pp. 278–453.

United Nations General Assembly (1982). Convention on the law of the sea.

Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science & business media.

Verma, B. (2009). Long range identification and tracking (LRIT) apropos global maritime security. *Maritime Affairs: Journal of the National Maritime Foundation of India*, vol. 5, no. 1, pp. 39–56.

Virjonen, P., Nevalainen, P., Pahikkala, T. and Heikkonen, J. (2018). Ship movement prediction using $k$-NN method. In: *2018 Baltic Geodetic Congress (BGC Geomatics)*, pp. 304–309.

Walker, B., Sander, G., Thompson, M., Burns, B., Fellerhoff, R. and Dubbert, D. (1996). A high-resolution, four-band SAR testbed with real-time image formation. In: *IGARSS'96. 1996 International Geoscience and Remote Sensing Symposium*, vol. 3, pp. 1881–1885. IEEE.

Wang, C., Ren, H. and Li, H. (2020). Vessel trajectory prediction based on AIS data and bidirectional GRU. In: *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, pp. 260–264.

Wang, S. and He, Z. (2021). A prediction model of vessel trajectory based on generative adversarial network. *Journal of Navigation*, vol. 74, no. 5, p. 1161–1171.

Wang, X., Mueen, A., Ding, H., Trajcevski, G., Scheuermann, P. and Keogh, E. (2013). Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309.

Wang, Y., Zhang, Z., Li, N., Hong, F., Fan, H. and Wang, X. (2017). Maritime surveillance with undersampled SAR. *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 8, pp. 1423–1427.

Welch, G., Bishop, G. *et al.* (1995). An introduction to the Kalman Filter.

Wijaya, W.M. and Nakamura, Y. (2013). Predicting ship behavior navigating through heavily trafficked fairways by analyzing AIS data on Apache HBase. In: *2013 First International Symposium on Computing and Networking*, pp. 220–226.

Xiao, Z., Fu, X., Zhang, L. and Goh, R.S.M. (2020). Traffic pattern mining and forecasting technologies in maritime traffic service networks: A comprehensive survey. *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1796–1825.

Xu, S., An, X., Qiao, X., Zhu, L. and Li, L. (2013). Multi-output least-squares support vector regression machines. *Pattern Recognition Letters*, vol. 34, no. 9, pp. 1078–1084.

Xu, X. (2020). Context-based trajectory prediction with LSTM networks. In: *2020 The 3rd International Conference on Computational Intelligence and Intelligent Systems*, CIIS 2020, p. 100–104. Association for Computing Machinery, New York, NY, USA.

Yang, D., Wu, L., Wang, S., Jia, H. and Li, K.X. (2019). How big data enriches maritime research – a critical review of automatic identification system (AIS) data applications. *Transport Reviews*, vol. 39, no. 6, pp. 755–773.

Yuan, Z., Liu, J., Liu, Y. and Li, Z. (2019). A novel approach for vessel trajectory reconstruction using AIS data. In: *The 29th International Ocean and Polar Engineering Conference*. OnePetro.

Zhang, C., Bin, J., Wang, W., Peng, X., Wang, R., Halldearn, R. and Liu, Z. (2020). AIS data driven general vessel destination prediction: A random forest based approach. *Transportation Research Part C: Emerging Technologies*, vol. 118, p. 102729.

Zhang, L., Liu, Z., Zou, R., Guo, J. and Liu, Y. (2014). A scalable CSMA and self-organizing TDMA MAC for IEEE 802.11 p/1609. x in VANETs. *Wireless Personal Communications*, vol. 74, no. 4, pp. 1197–1212.

Zhao, Z., Ji, K., Xing, X., Zou, H. and Zhou, S. (2014). Ship surveillance by integration of spaceborne SAR and AIS–review of current research. *The Journal of Navigation*, vol. 67, no. 1, pp. 177–189.

Zissis, D., Xidias, E.K. and Lekkas, D. (2015). A cloud based architecture capable of perceiving and predicting multiple vessel behaviour. *Applied Soft Computing*, vol. 35, pp. 652–661.

Zissis, D., Xidias, E.K. and Lekkas, D. (2016). Real-time vessel behavior prediction. *Evolving Systems*, vol. 7, no. 1, pp. 29–40.

# GLOSSARY

**anchored**

A action of a vessel being hold in place by its anchor, whilst floating on water.

**bathymetry**

The measurement of the depth of water in oceans, rivers, or lakes.

**berthing**

The action or process of mooring a ship in its allotted place.

**bow**

Refers to the most front part of a vessel, the part that usually first break waves at sea.

**caution area**

An area identified by VTS operators in which numerous vessel route intersections exist together with high traffic flow.

**Digital Selective Calling**

A standard for the transmission of predefined digital messages via different radio systems, such systems include the medium-, high-, and very-high frequency systems.

**Haversine distance**

The angular distance between two coordinates on the the surface of a sphere (earth in this thesis).

**heading**

The compass direction a vessel's bow is pointed at.

**moored**

When a vessel is secured by ropes, cables or an anchor to keep it in place, usually at ports and marinas.

**port**

A facility usually at shore, where vessels uploads or offloads cargo or passengers.

**stern**

Refers to most the back part of a vessel, where the propellers are.

# APPENDIX A

# THE DISCRETE KALMAN FILTER

The Kalman Filter (KF) can take on various forms, one of which is the so called discrete KF (DKF). The DKF (also known as the classic KF) can be applied to linear systems. The DKF is introduced in Chapter 4 as an AIS trajectory prediction algorithm. In this thesis, the terms KF and DKF are used interchangeably. The KF was originally proposed by Kalman (1960). The material presented in this appendix closely follows the content contained in Welch *et al.* (1995). An example from Russell and Norvig (2002) is also presented.

A KF allows for the continuous estimation of a state, continuously adjusting itself, as new observations are recorded. The inner workings of the DKF are discussed in the sections that follow. The in-depth presentation in the remaining sections of this Appendix should enable the reader to obtain a better understanding of the DKF. In short, the KF is a set of mathematical equations that provides an efficient recursive computational solution to the least-squares problem. The DKF can estimate past, present and future states, and it can do these estimations even when the precise nature of the modelled system is unknown.

## A.1 THE PROCESS TO BE ESTIMATED

The DKF allows for the estimation of the state $\boldsymbol{x} \in \Re^n$ of a discrete time controlled process governed by the linear stochastic difference equation (Welch *et al.*, 1995):

$$x_{k+1} = \boldsymbol{A}_k x_k + \boldsymbol{B} u_k + w_k, \tag{A.1}$$

with a measurement $\boldsymbol{z} \in \Re^m$ that is

$$z_k = \boldsymbol{H}_k x_k + v_k. \tag{A.2}$$

The random variables $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ represent the process and measurement noise respectively. Both $\boldsymbol{w}_k$ and $\boldsymbol{v}_k$ are independent, white and normally distributed ($\mathcal{N}$), i.e. :

$$p(\boldsymbol{w}) \sim \mathcal{N}(0, \boldsymbol{Q}), \tag{A.3}$$

$$p(\boldsymbol{v}) \sim \mathcal{N}(0, \boldsymbol{R}). \tag{A.4}$$

- Matrix $\boldsymbol{A}$ $(n \times n)$ in difference Equation A.1, relates the state at time step $k$ to the state at $k+1$, in the absence of either a driving function or process noise. Therefore, we refer to $\boldsymbol{A}$ as the transition matrix. The value of matrix $\boldsymbol{A}$ is dependent on the problem at hand.
- Matrix $\boldsymbol{B}$ $(n \times l)$ relates the control input $\boldsymbol{u} \in \Re^l$ to the state $\boldsymbol{x}$. Therefore, it is called the output matrix. Matrix $\boldsymbol{B}$ is also problem dependent.
- Matrix $\boldsymbol{H}$ with dimensions $(m \times n)$ in measurement Equation A.2, relates the state to the measurement $\boldsymbol{z}_k$, and as such is known as the transformation matrix. $\boldsymbol{H}$ should be chosen in such a way to minimise the second moment $E[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)^T] = \boldsymbol{P}_k$, where "E" refers to the expectation operator, and "$(\cdot)^T$" denotes the transpose operator

Note that $\hat{x}_{k+1}^-, \hat{x}_k$ and $u_k$ will not always be scalars. For the multivariate DKF case, they are vectors of size $n$.

## A.2 THE COMPUTATIONAL ORIGINS OF THE FILTER

Let $\hat{\boldsymbol{x}}_k^- \in \Re^n$ (note the superscript: "$-$") be defined as the *a priori* state estimate at step $k$, given the knowledge of the process prior to step $k$. Let $\hat{\boldsymbol{x}}_k \in \Re^n$ be the *a posteriori* state estimate at step $k$ given measurement $\boldsymbol{z}_k$.

Let the *a priori* and *a posteriori* estimate errors be defined by,

$$\boldsymbol{w}_k^- \equiv \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k^-,$$

and

$$\boldsymbol{e}_k \equiv \boldsymbol{x}_k - \hat{\boldsymbol{x}}_k.$$

The *a priori* estimate's error covariance is then defined by:

$$\boldsymbol{P}_k^- = E[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-T}], \tag{A.5}$$

and the *a posteriori* estimate's error covariance by:

$$\boldsymbol{P}_k = E[\boldsymbol{e}_k \boldsymbol{e}_k^T]. \tag{A.6}$$

During the derivation of the equations for the DKF, the first goal is to find an equation that computes an *a posteriori* state estimate $\hat{\boldsymbol{x}}_k$, as a linear combination of an *a priori* estimate $\hat{\boldsymbol{x}}_k^-$ and a weighted difference between an actual measurement $\boldsymbol{z}_k$ and a prediction of said measurement $\boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-$ (shown in Equation A.7 below). The probabilistic origin and justification of Equation A.7 can be found in Section A.3.

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^- + \boldsymbol{K}(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-) \tag{A.7}$$

The difference $(\boldsymbol{z}_k - \boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-)$ in Equation A.7 is referred to as the residual, also known as the measurement error. The residual reflects the difference between the actual measurement $\boldsymbol{z}_k$ and the measurement prediction $\boldsymbol{H}_k\hat{\boldsymbol{x}}_k^-$ at time step $k$. If the residual is equal to the zero means that the two are in complete agreement.

The matrix $\boldsymbol{K}$ $(n \times m)$ in Equation A.7 is referred to as the Kalman gain (blending factor) that minimises the *a posteriori* covariance in Equation A.6. It can be computed using the following steps:

1. Substitute Equation A.7 into the definition for $\boldsymbol{e}_k$ shown above,
2. substitute that into Equation A.6,
3. perform the indicated expectations,
4. take the derivative of the trace* of the result with respect to $\boldsymbol{K}$,
5. set the result equal to zero, and
6. finally solve for $\boldsymbol{K}$.

One possible solution for $\boldsymbol{K}$ which minimises Equation A.6 is given by Equation A.8. It is important to note, the KF equations can be manipulated into several forms, i.e. Equation A.8 is just one popular form of the Kalman gain and is the form which is used throughout the thesis.

$$\boldsymbol{K}_k = \boldsymbol{P}_k^-\boldsymbol{H}_k^T(\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} = \frac{\boldsymbol{P}_k^-\boldsymbol{H}_k^T}{\boldsymbol{H}_k\boldsymbol{P}_k^-\boldsymbol{H}_k^T + \boldsymbol{R}_k}\,. \tag{A.8}$$

Inspecting Equation A.8, we notice that as the measurement covariance $\boldsymbol{R}_k$ approaches zero, then the Kalman gain $\boldsymbol{K}$ weighs the residual more heavily. Specifically,

*The trace of a matrix is only defined for square matrices and is the sum of the elements on the diagonal, mathematically we express this as $tr(\boldsymbol{S}) = \sum_{i=1}^{N} \boldsymbol{S}_{ii}$

$$\lim_{\boldsymbol{R}_k \to 0} \boldsymbol{K}_k = \boldsymbol{H}_k^{-1} \ .$$

In other words, as the measurement error covariance $\boldsymbol{R}_k$ approaches zero, the actual measurement $\boldsymbol{z}_k$ is "trusted" more and more, while the predicted measurement $\boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-$ is trusted less and less.

As the *a priori* estimate error covariance $\boldsymbol{P}_k^-$ approaches zero, $\boldsymbol{K}$ weighs the residuals less and less. Specifically,

$$\lim_{\boldsymbol{P}_k^- \to 0} \boldsymbol{K}_k = 0 \ .$$

In other words, as the *a priori* estimate error covariance $\boldsymbol{P}_k^-$ approaches zero, the measurement $\boldsymbol{z}_k$ is trusted less and less, while the predicted measurement $\boldsymbol{H}_k \hat{\boldsymbol{x}}_k^-$ is trusted more and more.

## A.3 PROBABILISTIC ORIGINS OF THE FILTER

The justification of Equation A.7 is rooted in the probability of the *a priori* estimate $\hat{\boldsymbol{x}}_k^-$ conditioned on all prior measurements $\boldsymbol{z}_k$, i.e. in Bayes' rule.

We point out that the DKF maintains the first two moments of the state distribution, where the first moment reflects the mean and the second moment the variance. The moments are defined below:

$$E[\boldsymbol{x}_k] = \hat{\boldsymbol{x}}_k$$

$$E[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)^T] = \boldsymbol{P}_k.$$

The *a posteriori* estimate in Equation A.7 reflects the mean of the state distribution which is normally distributed if the conditions in Equations A.3 and A.4 have been met. The *a posteriori* estimate error covariance in Equation A.6 reflects the state of the distribution (Welch *et al.* (1995)). In other words:

$$\begin{aligned} p(\boldsymbol{x}_k|\boldsymbol{z}_k) &\sim \mathcal{N}(E[\boldsymbol{x}_k], E[(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_k)^T]) \\ &= \mathcal{N}(\hat{\boldsymbol{x}}_k, \boldsymbol{P}_k) \end{aligned}$$

The following resources has more on the probabilistic origins of the DKF, Maybeck (1982), Chen (1992) and Jacobs (1974).

Russell and Norvig (2002) contains a thorough explanation on the DKF. We summarise this section below.

### A.3.1 A simple one-dimensional KF example

In this section we present a simple one dimensional DKF example. This example should enable the reader to better understand the DKF itself and its probabilistic origins. The example presented here is a modification of the one presented in (Russell and Norvig, 2002).

The example below is a univariate example, with the general case being shown in the next section. Showcasing how the DKF is tied to the mathematical properties of Gaussian distributions.

Let us consider a temporal model which describes a random walk of a single continues state $x_k$ with a noisy observation $z_k$. For example $x_k$ could denote the "consumer confidence index" at time step $k$, which is measured by a random survey $z_k$.

Let the prior distribution of the state model with mean $\mu_0$ and variance $\sigma_0^2$ be equal to:

$$P(x_0) = \alpha\, e^{-\frac{1}{2}\left(\frac{(x_0-\mu_0)^2}{\sigma_0^2}\right)},$$

where $\alpha$ represents a normalising constant throughout this section (it greatly simplifies the equations). Moreover, assume that the state transition model adds a Gaussian perturbation of constant variance $\sigma_x^2$ , i.e

$$P(x_{k+1}|x_k) = \alpha\, e^{-\frac{1}{2}\left(\frac{(x_{k+1}-x_k)^2}{\sigma_x^2}\right)}. \tag{A.9}$$

Furthermore, let the measurement model be described by:

$$P(z_k|x_k) = \alpha\, e^{-\frac{1}{2}\left(\frac{(z_k-x_k)^2}{\sigma_z^2}\right)},$$

We can now compute the following:

$$\begin{aligned} P(x_1) &= \int_{-\infty}^{\infty} P(x_1|x_0)P(x_0)dx_0 \\ &= \alpha \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\frac{(x_1-x_0)^2}{\sigma_x^2}\right)} e^{-\frac{1}{2}\left(\frac{(x_0-\mu_0)^2}{\sigma_0^2}\right)} dx_0 \\ &= \alpha \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(\frac{\sigma_0^2(x_1-x_0)^2+\sigma_x^2(x_0-\mu_0)^2}{\sigma_0^2\sigma_x^2}\right)} dx_0\ . \end{aligned} \tag{A.10}$$

It should be noted that the exponent in the last equation of A.10 is the sum of two expressions which are quadratic in $x_0$ and hence itself quadratic in $x_0$. Using *completing of the square* allows us to rewrite any quadratic equation $ax_0^2 + bx_0 + c$ as a the sum of a squared term $a(x_0 - \frac{-b}{2a})^2$ and

residual term $c - \frac{b^2}{4a}$ which is independent of $x_0$ (Russell and Norvig (2002)). If we take the factor associated with the residual term out of the integral in A.10 we obtain:

$$P(x_1) = \alpha e^{-\frac{1}{2}\left(c-\frac{b^2}{4a}\right)} \int_{-\infty}^{\infty} e^{-\frac{1}{2}\left(a(x_0-\frac{-b}{2a})^2\right)} dx_0. \tag{A.11}$$

The integral factor of A.11, equates to one (it is the integral of a Gaussian curve over its full range). We are thus left with the factor associated with the residual term, which is quadratic in $x_1$. After some simplification we obtain:

$$P(x_1) = \alpha e^{-\frac{1}{2}\left(\frac{(x_1-\mu_0)^2}{\sigma_0^2+\sigma_x^2}\right)}. \tag{A.12}$$

To complete the update step of the DKF, we need to condition on the observation at the first time step $z_1$, i.e.

$$\begin{aligned} P(x_1|z_1) &= \alpha P(z_1|x_1)P(x_1) \\ &= \alpha e^{-\frac{1}{2}\left(\frac{(z_1-x_1)^2}{\sigma_z^2}\right)} e^{-\frac{1}{2}\left(\frac{(x_1-\mu_0)^2}{\sigma_0^2+\sigma_x^2}\right)} \end{aligned} \tag{A.13}$$

If we now combine the exponents in A.13 and then complete the square in the resulting exponent we find:

$$P(x_1|z_1) = \alpha e^{-\frac{1}{2}\left(\frac{(x_1-\frac{(\sigma_0^2+\sigma_x^2)z_1+\sigma_z^2\mu_0)^2}{\sigma_0^2+\sigma_x^2+\sigma_z^2}}{(\sigma_0^2+\sigma_x^2)\sigma_z^2/(\sigma_0^2+\sigma_x^2+\sigma_z^2)}\right)} \tag{A.14}$$

After one update cycle, a new Gaussian distribution is obtained for the state variable. In general, the new mean and standard deviation of the state variable can thus be computed from the old mean and standard deviation via the following equations:

$$\mu_{k+1} = \frac{(\sigma_k^2+\sigma_x^2)z_{k+1} + \sigma_z^2\mu_k}{\sigma_k^2+\sigma_x^2+\sigma_z^2}, \tag{A.15}$$

and

$$\sigma_{k+1}^2 = \frac{(\sigma_k^2+\sigma_x^2)\sigma_z^2}{(\sigma_k^2+\sigma_x^2+\sigma_z^2)}. \tag{A.16}$$

Looking at Figure A.1, we see that one update cycle is shown of the DKF for specific values of the

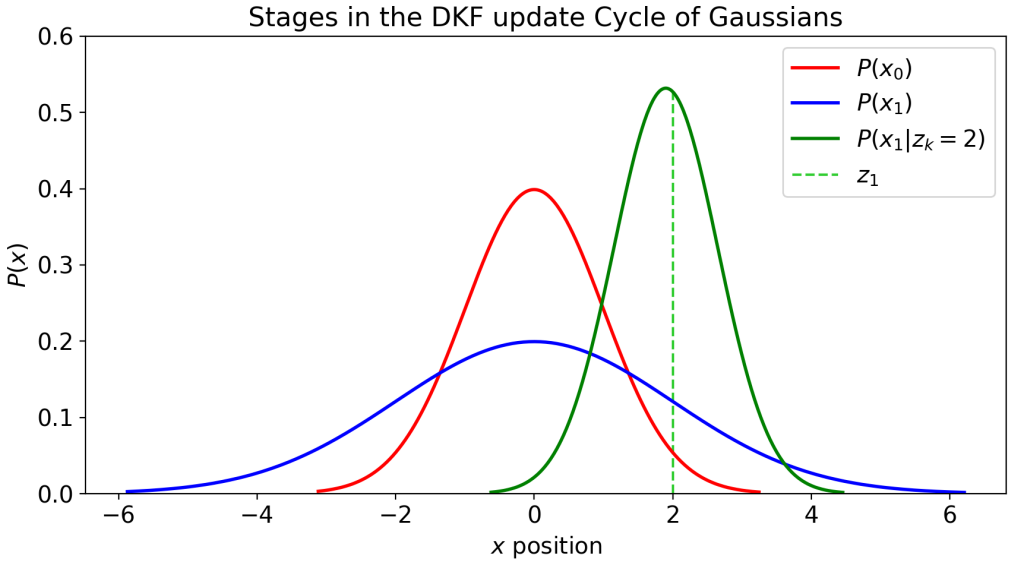


Figure A.1: The stages of the DKF update cycle are given in this example. The *a priori* state distribution is parameterised by $\mu_0 = 0.0$ and $\sigma_0 = 1$. The transition noise is given by $\sigma_x = 2$, while the measurement noise is given by $\sigma_z = 1$. The first observation is $z_1 = 2$ (marked by the dashed line). The prediction of $P(x_1)$ is flattened out, relative to $P(x_0)$ by the transition noise. The mean of the posterior distribution $P(x_1|z_1)$ is a bit more to the left of the observation $z_1$, due to the mean being a weighted average of the prediction and the observation.

transition and measurement models. Going through this example in more detail:

1. Let the calculation of the new mean $\mu_{k+1}$ be interpreted as a weighted mean of the new observation $z_k$ and old mean $\mu_k$. If the observation made is unreliable to the corresponding variance $\sigma_z^2$ will be large, the old mean would be weighted more. If the old mean is unreliable, where $\sigma_k^2$ is large, or where the process is unpredictable where $\sigma_x^2$ is large, more weight will be given to the observed observation.
2. The update of the variance $\sigma_{k+1}^2$ is independent of the observation observations, we can therefore compute the sequence of variance values in advance.
3. The sequence of variance values converges quickly to a fixed value, depending only on $\sigma_x^2$ and $\sigma_z^2$, thereby substantially simplifying the subsequent calculations.

## A.4 THE DKF ALGORITHM

In this section the Discrete Kalman Filter (DKF) algorithm is discussed at a high-level, for the multivariate use case.

The DKF estimates a process using feedback control: the filter estimates a state at a given time point and then obtains feedback in the form of measurements in the presence of noise. The two sets of equations that make up of the DKF is the predictor and measurement update equations.

The DKF has two sets of equations, predictor and measurement update equations. These two sets of equations are denoted in Figure A.2 and in Equations A.17 A.18, A.19, A.20 and A.21 below. The time update in Figure A.2 refers to the predicted state at a specific time step, and the measurement update refers to the corrector equations updating the underlying Gaussian distribution functions.

The predictor equations are responsible for projecting forward the current state and associated error covariance estimates, to obtain the *a priori* estimate at the next time step $k+1$. The measurement update equations are responsible for the feedback, incorporating the new measurement of the current state into the *a priori* estimate, resulting in an improved *a posteriori* estimate.The final algorithm resembles a *predictor-corrector* algorithm for solving numerical problems, as shown in Figure A.2 indicated by the arrows.

Equation A.17 and A.18 below denotes the predictor equations.

$$\hat{\boldsymbol{x}}_{k+1}^{-} = \boldsymbol{A}_k \hat{\boldsymbol{x}}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{A.17}$$

$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{A}_k \boldsymbol{P}_k \boldsymbol{A}_k^T + \boldsymbol{Q}_k \tag{A.18}$$

These equations project the state and covariance from time step $k$ to $k+1$. Matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}$ are from Equation A.1, while $\boldsymbol{Q}_k$ is from Equation A.3.

The measurement update equations are represented by Equations A.19, A.20, and A.21:

$$\boldsymbol{K}_k = \boldsymbol{P}_k^{-} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_k^{-} \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{A.19}$$

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k^{-} + \boldsymbol{K}(\boldsymbol{z}_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_k^{-}) \tag{A.20}$$

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{A.21}$$

The first objective during the measurement update, is to compute the Kalman gain $\boldsymbol{K}_k$. The process is then measured obtaining $\boldsymbol{z}_k$, and then the *a posteriori* state can be estimated by incorporating the measurement, as in Equation A.20. The final step is to obtain the *a posteriori* error covariance via Equation A.21 (Welch *et al.* (1995)). Note that Equation A.19 and A.20 is similar to Equation A.8 and A.7 respectively, they are repeated for completeness.

After each predictor and measurement update pair, the process is repeated where the previous *a posteriori* estimates are used to predict the new *a priori* estimates. The recursive nature of the DKF makes it suitable for practical implementations and more feasible compared to other methods such as the Weiner filter (Chen, 1992). The Weiner filter was designed to operate on all data directly for each estimate, where the KF recursively conditions the current state estimate on all past measurements, not having to fit a model on all the data at every iteration. Figure A.2 is a complete overview of the recursive nature of the KF and the equations belonging to the predictor and measurement updates are shown. Please note that Figure A.2 is similar to Figure 4.1 with different subscripts.

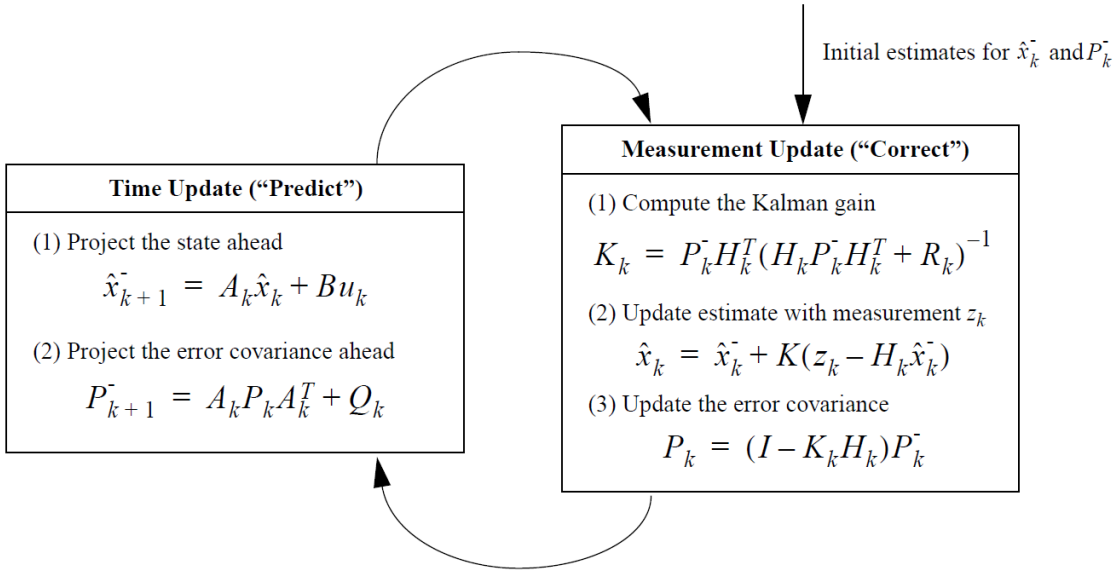


Figure A.2: The recursive Kalman filter operation with equations (Welch *et al.*, 1995).

## A.5 FILTER PARAMETERS AND PARAMETER TUNING

The process noise $\boldsymbol{Q}_k$ given by Equation A.3 and the measurement error covariance matrix $\boldsymbol{R}_k$ given by Equation A.4, can be measured prior to employing the DKF. In the case of the measurement error covariance $\boldsymbol{R}_k$ this makes sense, since we need to be able to measure the process while implementing the DKF. One can measure the process and get an initial estimate of the error covariance, by taking some off-line sample measurements to determine the variance of the measurement error.

In the case of the process noise $\boldsymbol{Q}_k$ the choice is more often than not, less deterministic. As an example, the noise source is often used to represent uncertainty in the process model given by Equation A.1. Sometimes a poor model can be used, by simply inserting enough uncertainty via the selection of $\boldsymbol{Q}_k$. In this case one would hope that the measurements of the process is reliable.

Whether or not we have a rational basis for choosing the parameters, often times superior filter performance (from a statistical point of view) can be obtained by tuning the parameters $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$. The tuning process is usually an off-line procedure and can be done by implementing a DKF on the off-line observations.

Under the conditions where $\boldsymbol{Q}_k$ and $\boldsymbol{R}_k$ are constant, both the estimation error $\boldsymbol{P}_k$ and the Kalman gain $\boldsymbol{K}_k$ will stabilise quickly and remain constant. If it is the case that they remain constant, these parameters can be pre-computed by running the filter off-line, or for example solving Equation A.18 for a steady state value of $\boldsymbol{P}_k$, by defining $\boldsymbol{P}_k^- \equiv \boldsymbol{P}_k$ and solving for $\boldsymbol{P}_k$.

It is very common that the measurement error does not remain a constant. For example, when receiving signals from vessels to AIS receivers, vessels closer to the receiver will have less noise compared to the vessels further away from the receiver. The process noise $\boldsymbol{Q}_k$, sometimes changes dynamically during the operations of the DKF in order to adjust to different dynamics. For example, when we are tracking a vessel, we might lower the magnitude of $\boldsymbol{Q}_k$ if the vessel is moving slowly, and increase its magnitude if the dynamics starts changing rapidly. In such a case, $\boldsymbol{Q}_k$ can be used not only to model the uncertainty of the underlying model, but also the uncertainty of the vessel's movement.

## A.6 A SIMPLE 1-D REAL-WORD DKF EXAMPLE

We now show an example of a DKF tracking the location of a constant velocity speed boat model over time. The speedboat is moving away from the harbour and we are able to measure the location of the boat in discrete time intervals of 0.2s. Let the underlying real trajectory of the speedboat with a constant velocity be denoted by $y = \frac{3}{20}\left(t^2 - 3t\right)$.

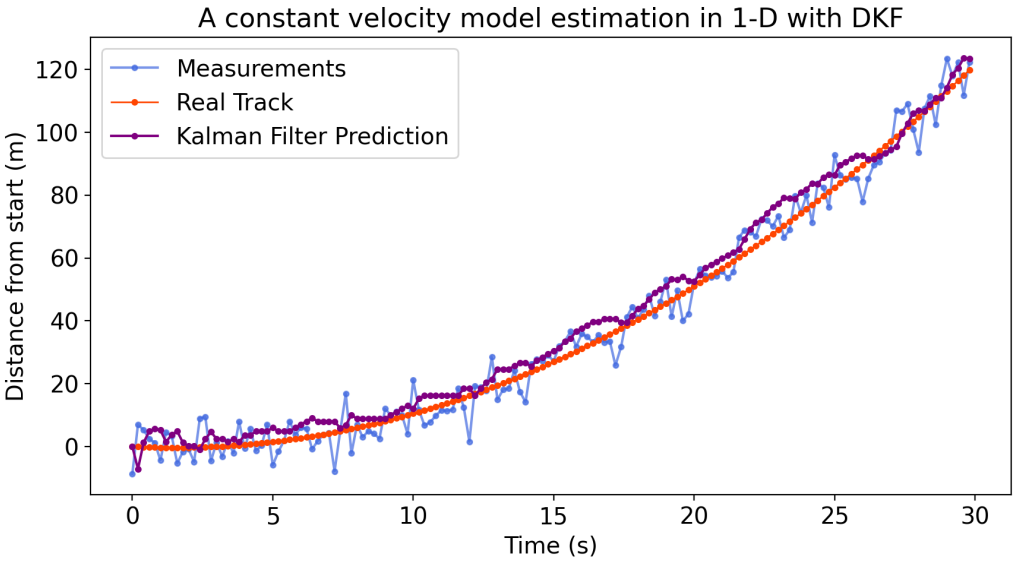


Figure A.3: A constant velocity speedboat model in one-dimension (1-D) and the DKF tracking over time

Investigating Figure A.3, we see the the measurements of the speedboat's location is denoted in blue (in reality Gaussian noise was added to the true model to simulate sensor noise). The true movement model (denoted in red) is also shown, from which the noisy measurement were sampled from.

Looking at the purple line we see the ability of the DKF to track the location of the speedboat in the presence of noise, closely modelling the underlying function. We see that the DKF prediction for the first 2.6s is slightly erratic, but starts to be less erratic as the corresponding error covariance matrices start to converge and the underlying function is being modelled in the presence of noise.

# APPENDIX B

# LINEAR REGRESSION MODEL

The Linear Regression Model (LRM) presented in this appendix is a summary of Chapter 3 from the book "An Introduction to Statistical Learning" by James *et al.* (2013).

The purpose of this appendix is to give a short overview of the LRM and its origins to equip the reader with enough information to understand the LRM (see Section 4.3) and LRMAC (see Section 4.5).

## B.1 SIMPLE LINEAR REGRESSION

Linear regression is a method that allows for the prediction of a numerical value (qualitative variable). The predicted value is referred to as the response variable, on the basis of a predictor variable. Let $Y$ denote the response variable and $X$ denote the predictor variable.

Let the LRM be mathematically denoted by:

$$Y \approx \beta_0 + \beta_1 X \tag{B.1}$$

where $\approx$ denotes the approximation of Y given the regression coefficients $\beta_0$ and $\beta_1$ and predictor $X$.

The regression coefficients $\beta_0$ and $\beta_1$ represents the intercept and slope of the linear model respectively. The values of $\beta_0$ and $\beta_1$ are unknown and estimated based on the problem at hand.

The LRM is trained (estimated) by a set of recorded predictors and response variables, where the regression coefficients $\beta_0$ and $\beta_1$ are estimated. The LRM can be classified as a supervised machine learning method, that obtains the best linear fit (relationship) between the predictors and the response variables as found by some measure.

Let the trained LRM be denoted by:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{B.2}$$

where $\hat{}$ ("hat") denotes the estimated value of the response and coefficients. Let $\hat{y}$ indicate the prediction of $Y$ when $X = x$.

### B.1.1 Example Problem

Figure B.1 depicts the test score of 35 students. The figure reflects the score obtained for their mathematics examination versus the total time they spent studying for the test. In reality, the observations (in teal) were generated from $y = \frac{5}{3}x + 10$ with random white Gaussian noise added. The true model is denoted in red and the estimated LRM denoted in purple.

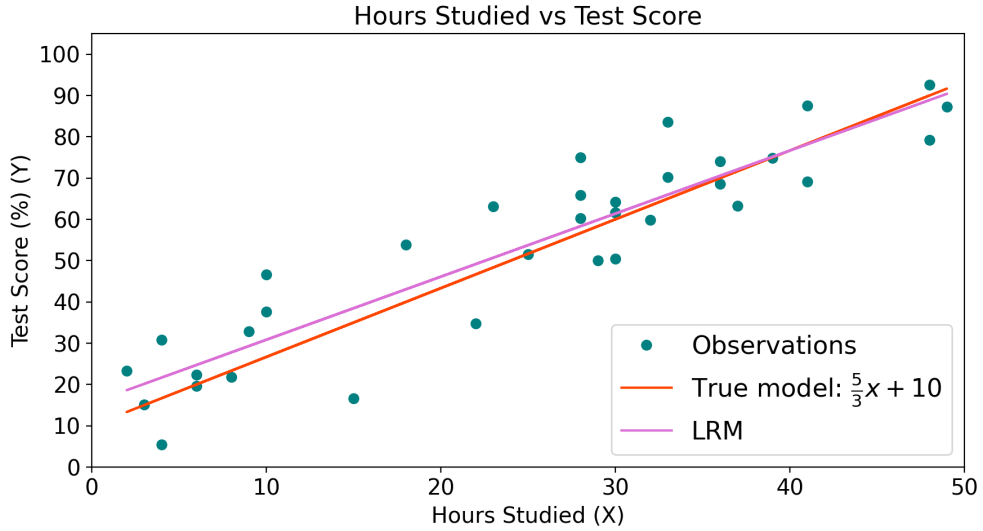


Figure B.1: A hypothetical problem. The amount of time 35 learners studied for a test and the corresponding test score each student obtained is depicted.

Throughout the remainder of this appendix we refer back to this problem set, as the LRM is explained.

### B.1.2 Coefficient Estimation

In order to create a functioning LRM we have to estimate the coefficients $\beta_0$ and $\beta_1$. Let the mathematical representation of the values present in Figure B.1 be denoted by

$$(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_n, y_n), \tag{B.3}$$

where $n = 35$ for this particular problem. The sequence represents the recorded observation pairs of the predictor and response variables.

The goal of the LRM is to estimate the coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$ in such a way that fits the data well, closely modelling the underlying function of the data. The goal is to find the values of the intercept $\hat{\beta}_0$ and slope $\hat{\beta}_1$ such that the resulting line is as close as possible to all the $n = 35$ data points. The most common approach to estimate the regression coefficients involves minimising the least squares criterion.

### B.1.3 Least Squares Minimisation

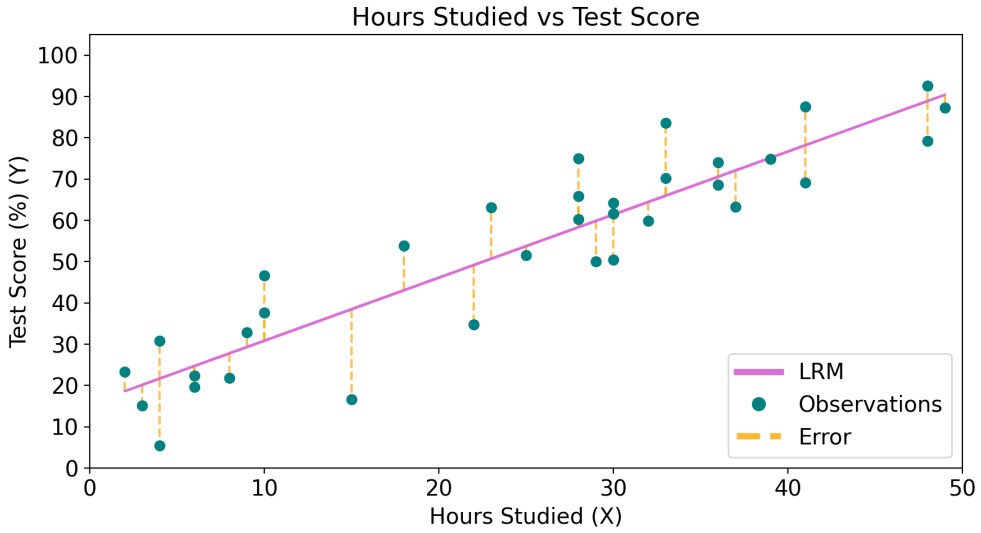


Figure B.2: Least Squares visualisation of the LRM

In short, the minimisation of the Least-Squares (LS) criterion can be best explained by making reference to Figure B.2, which illustrates the notion of minimising the error (distance) from the observed observations to the fitted line.

Let $y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for $Y$ based on the $i^{\text{th}}$ value of $X$. Let the $i^{\text{th}}$ error (also known as the residual) $e_i$ be denoted by $e_i = y_i - \hat{y}_i$, the difference between the predicted observation by the LRM at $x$ and the observed observation at $x$. Let the residual sum of squares (RSS) be defined by:

$$\text{RSS} = e_1^2 + e_2^2 + ... + e_n^2, \tag{B.4}$$

which can also be written as:

$$\text{RSS} = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + (y_2 - \hat{\beta}_0 - \hat{\beta}_1 x_2)^2 + ... + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2. \tag{B.5}$$

The LS approach chooses the values of $\hat{\beta}_0$ and $\hat{\beta}_1$ in such a way to minimise the RSS. The minimisers are defended below:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \tag{B.6}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}, \tag{B.7}$$

where both $\bar{y} = \sum_{i=1}^{n} y_i$ and $\bar{x} = \sum_{i=1}^{n} x_i$ represent the respective sample means. Equations B.6 and B.7 defines the LS coefficient estimates for the simple LRM. A derivation of these coefficients can be seen in Chapter 14 of Rice (2006).

The values of $\hat{\beta}_0$ and $\hat{\beta}_1$ resulting in the smallest RSS will be the ideal set of coefficients as determined by the LS estimate. We want to fit a model where the LRM in B.2 has the the smallest overall distance to all the observations. The LS minimises the distance between the fitted line and the all the observations. Therefore, the result of the LS will be one where the error/distance between the observations the fitted line are the smallest, resulting in the best linear fit (the error is denoted in orange in Figure B.2, and the sum of these errors are minimised by the LS).

Given the example in Figure B.1, the LRM coefficient pair that resulted in the best fit (smallest error) using LS were $\hat{\beta}_0 = 15.559$ and $\hat{\beta}_1 = 1.528$ with an associated error of 43.173.

Figure B.3, contains a 3 dimensional surface of the errors associated with each coefficient pair. With the knowledge that the pair with the smallest error will result in the best fit the goal will be to find the minimum value of the surface in the $z$-axis (lowest point).

The lowest point of the surface in Figure B.3 is represented by a red dot. The coordinates of the dot is equal to that of the coefficients and the calculated error from the LRM.

In Rice (2006) and James *et al.* (2013), more information on the model assumptions and measures for the quality of a fit for the LRM can found.

The LRM fitted throughout this thesis, including in the LRMAC, was done with the help of scikit-

learn's linear regression model* (Pedregosa *et al.*, 2011).

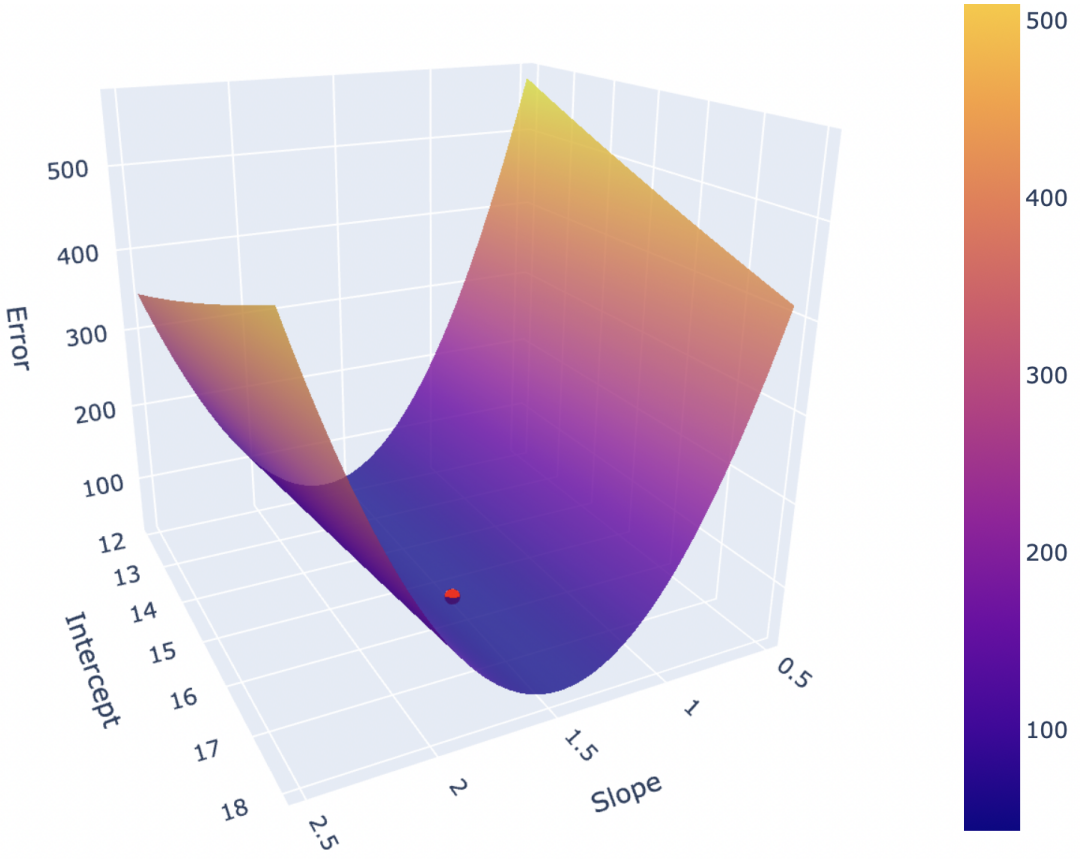


Figure B.3: Loss Function visualisation for different coefficient combinations $\hat{\beta}_0$ and $\hat{\beta}_1$ with the associated errors

*Scikit-learn is a popular ML library for the Python programming language. The library has various supervised and unsupervised learning models.

# APPENDIX C

# TRAJECTORIES USED

In this appendix, the datasets used to carry out the experiments are listed.

In Table C.1 below*, we denote the linear trajectories that the DKF and LRM were tested on to compare both methods. We denote the subsets that were linear from the paths, as both the LRM and DKF are linear methods. The vessels and data can be found in the dataset published by Ray *et al.* (2019).

Table C.1: Vessel extracted trajectory statistics for the LRM and DKF comparison

| MMSI | Number of Observations | Total time (seconds) | Minimum Longitude (UTM) | Maximum Longitude (UTM) | Minimum Latitude (UTM) | Maximum Latitude (UTM) | Trajectory from Observation | Trajectory to Observation |
|---|---|---|---|---|---|---|---|---|
| 538004506 | 339 | 3749 | 357377.938 | 372795.416 | 5343054.260 | 5350213.777 | 10 | 350 |
| 226105000 | 399 | 4661 | 347730.771 | 367452.230 | 5322232.509 | 5342445.685 | 420 | 850 |
| 235097013 | 199 | 2000 | 366276.749 | 369020.055 | 5347739.500 | 5355559.711 | 200 | 400 |
| 305714000 | 594 | 9269 | 348355.850 | 365699.479 | 5332172.500 | 5345131.707 | 5 | 600 |
| 304519000 | 449 | 4510 | 351732.188 | 373527.179 | 5345997.028 | 5350864.895 | 300 | 750 |
| 304927000 | 449 | 7269 | 348404.971 | 383371.404 | 5337055.240 | 5355671.446 | 300 | 750 |
| 636092331 | 491 | 5159 | 353566.873 | 365997.282 | 5333301.686 | 5344310.035 | 4358 | 4850 |
| 224389000 | 246 | 2592 | 364813.604 | 375080.329 | 5332247.399 | 5334147.887 | 128 | 375 |
| 220540000 | 439 | 4399 | 353023.745 | 377274.679 | 5330971.559 | 5334506.093 | 120 | 560 |
| 211286440 | 589 | 6097 | 354106.758 | 381920.651 | 5343457.627 | 5354947.954 | 2210 | 2800 |
| 207138000 | 439 | 4430 | 370761.799 | 391446.494 | 5349534.776 | 5358391.650 | 150 | 590 |
| 207138000 | 409 | 4100 | 370572.790 | 391001.242 | 5331631.927 | 5340167.846 | 790 | 1200 |
| 215901000 | 549 | 5921 | 354645.501 | 378062.302 | 5343001.012 | 5353829.393 | 50 | 600 |
| 227146400 | 539 | 6240 | 351890.805 | 379493.479 | 5332266.859 | 5346617.394 | 960 | 1500 |
| 314237000 | 499 | 7580 | 361252.193 | 387443.920 | 5344364.725 | 5357101.506 | 1000 | 1500 |
| 565494000 | 529 | 5779 | 349407.899 | 381515.888 | 5340629.108 | 5353520.956 | 20 | 550 |
| 244740921 | 398 | 4558 | 366109.231 | 382399.802 | 5349243.974 | 5355239.578 | 5200 | 5600 |
| 227330000 | 499 | 5301 | 358668.372 | 378879.619 | 5332689.010 | 5345856.820 | 50 | 550 |
| 227372000 | 409 | 6800 | 346026.235 | 378985.611 | 5331207.008 | 5339311.246 | 10 | 420 |
| 228130000 | 439 | 5469 | 356236.883 | 374059.358 | 5315182.721 | 5333469.559 | 20 | 450 |
| 228727000 | 399 | 3802 | 361453.774 | 383768.460 | 5347210.130 | 5355820.092 | 100 | 500 |
| 227988000 | 399 | 4070 | 373132.546 | 390953.780 | 5353627.921 | 5358073.197 | 1600 | 2000 |
| 244925000 | 449 | 5862 | 347661.989 | 372561.225 | 5335426.938 | 5351905.947 | 0 | 450 |
| 247224200 | 409 | 7117 | 340494.880 | 379971.192 | 5330172.298 | 5352693.834 | 210 | 620 |
| 249104000 | 449 | 4690 | 357631.456 | 377990.623 | 5346277.881 | 5352193.874 | 200 | 650 |
| 518866000 | 449 | 4692 | 358539.516 | 379077.364 | 5330651.551 | 5336750.802 | 350 | 800 |
| 276700000 | 399 | 4370 | 357459.972 | 374261.397 | 5342192.870 | 5350238.688 | 300 | 700 |
| 577228000 | 449 | 5239 | 367031.244 | 373072.270 | 5331947.203 | 5339918.516 | 300 | 750 |
| 227558000 | 449 | 4630 | 361334.439 | 375995.696 | 5331412.653 | 5338926.491 | 350 | 800 |
| 227364000 | 354 | 4250 | 355706.086 | 370294.428 | 5320749.824 | 5333984.934 | 495 | 850 |

In Table C.2 below, the list of vessel MMSIs are shown. The table shows the sub-trajectories used from Ray *et al.* (2019) for the experiments in this thesis. For replication purposes regarding the starting and end observation columns, assume that data for each vessel are sorted from the oldest

*Note the trajectories belong to UTM zone 30U

date-time observation to the newest. E.g. Regarding vessel MMSI 220503000, observation 0's associated timestamp (counting starts at 0) is older than the timestamp at observation at 99. The observation starting point is inclusive and the end point exclusive. The extracted sub-trajectories belong to the highways denoted in the SM in Figure 3.9, where N to S means that a vessel was travelling from the North to the South ("downward") and S to N the opposite. The vessel type and the total observational length (in seconds) is also shown for each MMSI.

| MMSI | Starting Observation | End Observation | # Observations | Total Time (seconds) | Vessel Type | Travelling Direction |
|---|---|---|---|---|---|---|
| 220503000 | 0 | 100 | 100 | 25836 | Cargo | S to N |
| 220603000 | 0 | 154 | 154 | 35428 | Tanker | S to N |
| 228337700 | 0 | 118 | 118 | 37219 | Tanker | S to N |
| 229605000 | 55 | 400 | 345 | 25231 | Cargo | N to S |
| 235080328 | 0 | 250 | 250 | 28492 | Cargo | N to S |
| 235084729 | 0 | 250 | 250 | 36295 | Cargo | N to S |
| 235094794 | 0 | 85 | 85 | 42313 | Tanker | S to N |
| 236339000 | 310 | 375 | 65 | 11703 | Cargo | S to N |
| 236386000 | 0 | 150 | 150 | 44124 | Cargo | S to N |
| 236481000 | 2 | 220 | 218 | 32404 | Cargo | N to S |
| 236668000 | 2 | 620 | 618 | 53707 | Tanker | S to N |
| 240411000 | 0 | 143 | 143 | 29867 | Tanker | S to N |
| 241066000 | 0 | 300 | 300 | 42584 | Cargo | S to N |
| 244424000 | 90 | 350 | 260 | 32697 | Cargo | S to N |
| 244703000 | 0 | 95 | 95 | 39922 | Cargo | N to S |
| 247078800 | 450 | 1050 | 600 | 47399 | Cargo | N to S |
| 248689000 | 0 | 200 | 200 | 83114 | Cargo | N to S |
| 249017000 | 100 | 380 | 280 | 26867 | Cargo | S to N |
| 249622000 | 5 | 200 | 195 | 45589 | Cargo | N to S |
| 249957000 | 160 | 370 | 210 | 45615 | Tanker | N to S |
| 250000963 | 0 | 150 | 150 | 32721 | Cargo | N to S |
| 255804890 | 0 | 667 | 667 | 16719 | Tanker | S to N |
| 256582000 | 0 | 300 | 300 | 58547 | Cargo | N to S |
| 256891000 | 350 | 800 | 450 | 47088 | Cargo | S to N |
| 256934000 | 0 | 40 | 40 | 21920 | Tanker | N to S |
| 258977000 | 0 | 210 | 210 | 38364 | Cargo | S to N |
| 271040029 | 0 | 52 | 52 | 39280 | Tanker | S to N |
| 271040594 | 3 | 160 | 157 | 57892 | Cargo | N to S |
| 271043873 | 0 | 197 | 197 | 39939 | Tanker | S to N |
| 275457000 | 740 | 880 | 140 | 39567 | Cargo | N to S |
| 304057000 | 0 | 215 | 215 | 97502 | Cargo | N to S |
| 304805000 | 263 | 450 | 187 | 39584 | Cargo | N to S |
| 304924000 | 0 | 250 | 250 | 123307 | Cargo | S to N |
| 319025300 | 1 | 110 | 109 | 47086 | Tanker | S to N |
| 319541000 | 0 | 43 | 43 | 18001 | Tanker | N to S |
| 353952000 | 0 | 59 | 59 | 43211 | Tanker | S to N |
| 419689000 | 0 | 79 | 79 | 50999 | Tanker | S to N |
| 565407000 | 0 | 97 | 97 | 38281 | Tanker | S to N |
| 566030000 | 0 | 450 | 450 | 57936 | Tanker | N to S |
| 636014352 | 0 | 205 | 205 | 61239 | Tanker | N to S |

Table C.2: All the vessel MMSIs used to evaluate the performance of the LRM, LRMAC and SPNS.