Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Fridrik Hilmar Zimsen Fridriksson

Bayesian Bridging Distribution Analysis for Model Based Intent and Long-term Prediction of Vessels

Master's thesis in Industrial Cybernetics
Supervisor: Edmund Førland Brekke
Co-supervisor: Trym Tengesdal

March 2021

NTNU
Norwegian University of
Science and Technology

Fridrik Hilmar Zimsen Fridriksson

Bayesian Bridging Distribution Analysis for Model Based Intent and Long-term Prediction of Vessels

Master's thesis in Industrial Cybernetics
Supervisor: Edmund Førland Brekke
Co-supervisor: Trym Tengesdal
March 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Preface

This thesis marks the end of my master's study in Industrial Cybernetics at NTNU, the Norwegian University of Science and Technology. The past few years have been challenging at times but extremely educational and rewarding.

I would like to give great thanks to my supervisor Edmund F. Brekke and my co-supervisor Trym Tengesdal for their support throughout this work, their great feedback and informative discussions. I also want to thank Ravinder Praveen K. Jain for his important insight to the problem ahead. Last but not least, I would like to thank Herbjörg Andrésdóttir my wife for her loving support and for helping me staying motivated throughout this work.

19/03/2021
Fridrik Hilmar Zimsen Fridriksson

Abstract

To ensure that autonomous surface vessels are safe for operation amongst the already existing traffic, it must be equipped with a robust collision avoidance (COLAV) system. An important part of such system is the ability to accurately detect potential threat and to avoid it. To achieve this a tracking model must be able to reliably predict future trajectory of other vessels that pose a potential threat.

For model-driven methods, the knowledge about the intent of other vessels can greatly increase how far into the future their trajectories can accurately be predicted. In this thesis, three model-driven methods for intent predictions have been analysed using measurements from Automatic Identification System (AIS). These methods are all based on Bayesian bridging distributions that compute the likelihood of possible destinations using the vessel's measurements as input. These are, (1) a Bayesian filtering approach that *bridges* the current state with the final state via a joint state of the two, (2) a Bayesian filtering approach that *bridges* the current state with the final state via a pseudo-measurement, and (3) a Bayesian smoothing approach that also uses a pseudo-measurement for *bridging*.

Three *bridged* motion models were considered for tracking, a constant velocity (CV) model, an equilibrium reverting velocity (ERV) model, and an Ornstein-Uhlenbeck (OU) model. These were then used with the different bridging methods for destination inference and future predictions. Good results were observed for most cases in predicting the intended destination. For predicting future states, both filtering approaches had good estimates up to about 15-30 min of future predictions, depending on the situation. Whereas the smoothing approach did not perform well in predicting future states.

Sammendrag

For å sørge for at autonome overflatefartøyer er sikre for bruk blant den allerede eksisterende trafikk, da må den være utstyrt med et antikollisjonssystem. En viktig del av et slikt system er evnen til å kunne oppdage potensielle trusler og unngå dem. For å oppnå dette må en målfølgingsmodell kunne predikere banen til andre overlatefartøyer.

For modellbaserte metoder, kan kunnskapen om intensjoner til andre fartøyer gi forbedrede prediksjoner om fremtidige tilstander. Denne masteroppgaven analyserer tre modellbaserte metoder for å predikere intensjoner til fartøyer ved bruk av reelle måledata fra Automatic Identification System (AIS). Disse metodene baserer seg på Bayesian bridging distribusjoner som beregner sannsynligheten for mulige destinasjoner og bruker fartøyets målinger som input. Disse er, (1) Bayesian filtreringsmetode som kobler nåværende tilstand med den endelige tilstanden via en joint tilstand som innholder begge tilstandene, (2) Bayesian filtreringsmetode som kobler nåværende tilstand med den endelige tilstanden via en pseudo-måling, og (3) Bayesian glattingsmetode som også bruker en pseudo-måling for koblingen.

Tre målfølgingsmodeller ble brukt, constant velocity (CV) modell, equilibrium reverting velocity (ERV) modell, og Ornstein-Uhlenbeck (OU) modell. Disse ble brukt i kombinasjon med Bayesian bridging metodene for destinasjonsprediksjoner og fremtidige tilstandsprediksjoner. Det ble observert gode resultater for destinasjonsprediksjonene ved bruk av de fleste modeller testet. For fremtidige tilstandsprediksjoner, begge filtrerings metodene hadde godt estimat opptil omtrent 15-30 min, varierende for forskjellige situasjoner. Derimot hadde glatting metoden dårlige tilstandsprediksjoner.

Table of contents

Preface i

Abstract ii

Sammendrag iii

Acronyms vi

Nomenclature vii

1 Introduction 1

1.1 Motivation 1

1.2 Problem description 2

1.3 Outline of thesis 3

2 Background material 4

2.1 Theorems and definitions 4

2.2 Stochastic state space models 6

2.3 Filtering and Smoothing 6

2.3.1 Bayesian filtering 6

2.3.2 Bayesian smoothing 7

2.3.3 Linear filtering and smoothing 8

2.4 Marine vessels in motion 11

3 Destination prediction using bridging distribution 13

3.1 Systems general tracking model 13

3.2 Bayesian distribution 14

3.2.1 Arrival time distribution 14

3.2.2 Destination inference 15

3.2.3 Arrival time distribution at any destination 16

3.3 Bridging model using joint state 16

3.3.1 State estimate and trajectory 17

3.4 Bridging model using Bayesian filtering 19

3.5 Bridging model using Bayesian smoothing 21

4 **Motion models** 26

4.1 Constant velocity model 26

4.2 Equilibrium reverting velocity 27

4.3 Ornstein-Uhlenbeck model 28

5 **Test scheme** 30

5.1 Scenarios 30

5.2 Using AIS measurements 31

5.3 State and measurement vectors 32

5.3.1 Tuning of covariance 34

6 **Results** 36

6.1 Model cases 36

6.2 Destination inference 36

6.2.1 *NE-S1*: Destination inference for a boat travelling from *NE* to *S1* 37

6.2.2 *W1-S1*: Destination inference for a ship travelling from *W1* to *S1* 40

6.2.3 *W1-E2*: Destination inference for a ship travelling from *W1* to *E2* 43

6.2.4 *N2-S2*: Destination inference for a ship travelling from *N2* to *S2* 45

6.2.5 *NE-W1*: Destination inference for a ship travelling from *NE* to *W1* 45

6.2.6 The difference between using the alternative mean and the given mean 47

6.2.7 The importance of correctly tuned model 49

6.2.8 Discussion, destination inference 51

6.3 Future predictions 51

6.3.1 *NE-S1*: Future predictions for a boat travelling from *NE* to *S1* 52

6.3.2 *W1-S1*: Future predictions for a ship travelling from *W1* to *S1* 52

6.3.3 *W1-E2*: Future predictions for a ship travelling from *W1* to *E2* 55

6.3.4 *NE-W1*: Future predictions for a ship travelling from *NE* to *W1* 56

6.3.5 Discussion, future predictions 56

7 **Conclusion** 59

8 **Suggestions for future work** 61

References 62

Appendices 64

A.I Examples of logarithmic representation 64

A.II Article's vs. Alternative mean: Destinations weights 65

A.III Predicted trajectory 67

Acronyms

AIS	Automatic Identification System
ASV	Autonomous Surface Vessels
BF	Bayesian filtering
BF1	Bayesian filtering equations using the current state vector
BF2	Bayesian filtering equations using the joint state vector of both the current and the final state
BS	Bayesian smoothing
COG	Course over Ground
COLAV	Collision avoidance
CV	Constant Velocity
ERV	Equilibrium Reverting Velocity
GNSS	Global Navigation Satellite System
KF	Kalman filter
MAP	Maximum a Posteriori
NEES	Normalised Estimation Error Squared
NIS	Normalised Innovation Squared
OU	Ornstein-Uhlenbeck
PED	Prediction Error Decomposition
RTS	Rauch-Tung-Striebel (smoother)
SDE	Stochastic Differential Equation
SOG	Speed over Ground

Nomenclature

λ	Latitude
$\mathbf{Q}$	State's covariance matrix
$\mathbf{R}$	Measurement's covariance matrix
$\mathbf{x}^a$	State vector including position and their derivatives, $[x, y, \dot{x}, \dot{y}]$
$\mathbf{x}^b$	State vector including position, course angle and their derivatives, $[x, y, \psi, \dot{x}, \dot{y}, \dot{\psi}]$
$\mathbf{x}_d$	The final state at destination d
$\mathbf{x}_f$	State vector at time t_f, the final state
$\mathbf{x}_k$	State vector at time t_k
$\mathbf{y}_k$	Measurement vector at time t_k
ψ	Course angle
σ_a	Process variance for acceleration
σ_h	Measurement variance for heading angle
σ_p	Measurement variance for position
σ_v	Measurement variance for velocity
σ_ψ	Process variance for course angle
$\tilde{\mathbf{y}}_f^{\{n\}}$	Pseudo-measurement at destination d_n
φ	Longitude
T	Time step k, i.e. $t_k - t_{k-1}$
t_*	Future predicted time, i.e. how far into the future a prediction has been made
t_k	Time at time step k, also used to represent the current time

Chapter 1

Introduction

1.1 Motivation

Safety on sea is of great importance to all involved in maritime activities to prevent both loss of life and economical loss. Unfortunately, accidents do take place and it is estimated that about 75% of accidents on sea are due to human errors [1]. Hence, by reducing human intervention, or removing it all together, could lead to improved safety on sea. Additionally, this reduces the amount of personnel needed onboard, leading to fewer people being subjected to risks and creating more space for cargo.

As the autonomous technology is rapidly being developed and improved, it will not be long until autonomous surface vessels (ASVs) will be a common way of transportation in marine environments. To be used among the already existing traffic, the ASVs must be proven to be safe for both themselves and other vessels. To get an approval for commercial use, it must demonstrate that its design holds an equivalent level of safety compared with already approved vessels of similar design [2]. To achieve this the ASVs must be equipped with a collision avoidance (COLAV) system that detects obstacles and determines if and how to avoid possible collision. For stationary obstacles, one can simply maneuver around the obstacle diverging slightly from the originally planned trajectory. For moving obstacles, however, COLAV can be more challenging as the motion is usually unknown and has to be estimated to then be able to predict the intended trajectory [3]. A common practice for estimating future trajectory is to assume the obstacle moves with a constant velocity while also keeping its course angle constant. This can be a decent estimation for a short-term prediction, but as the prediction goes further into the future the uncertainty increases quickly.

For long-term predictions, one could use data-driven methods where e.g. historical Automatic Identification System (AIS) data is used to predict the future trajectory. However, this is dependent on having access to enough data points to be accurate, and can be computationally heavy [4]. Another way is to use model-driven methods where the predicted trajectory is computed based on circumstances. For instance, taking into account previous measurements of a tracked object and possible destinations, e.g. harbours or ports, one could estimate the most likely trajectory. This can be achieved by *bridging* the states of the tracked object and the possible destinations by the means of Bayesian probability distributions [5, 6]. Furthermore, a model-driven method could be used in combination with a data-driven method, or in situations where insufficient amounts of historical data is available.

1.2 Problem description

Ahmad et al. [5] introduced a method of intent predictions for a tracked object using bridging distributions, and in [6] a different approach to the bridging method was introduced. Using Bayesian framework, the proposed algorithms compute the intended destination likelihood for an object in motion by utilising the available measurements. The models are low in complexity, require minimal training, and computations can be performed in parallel. This leads to an efficient intent inference with low computational cost. Additionally, the same framework can bu utilised to predict future states, e.g. position and velocity, with little additional cost since many parameter estimations have already been established.

In both articles, synthetic trajectories were generated for a maritime vessel travelling at sea towards one of six possible destinations that were positioned in an arc along a coastline. The trajectories were generated with a bridged constant velocity model, and even though the vessel will go towards a certain destination, the path it takes is random. This might be an unrealistic representation of the movement of maritime vessels in reality.

This thesis uses the work done in a preliminary specialisation project [7] as a basis. In [7] the bridging method introduced in [5] was analysed, where a similar simulation study was performed. It resulted in similar findings as reported in the article, that is, the bridging distribution resulted in good destination predictions for the simulated scenario. Additionally, the effect of *bridging* on the predictions was studied, which showed that it had better predictions with less uncertainty. The aim of this thesis is to continue the analysis of this bridging method, in addition to the bridging method introduced in [6], using real AIS measurements. The research questions to be answered are:

- *How will the different bridging approaches perform using real data measurements with regards to intent inference, as well as future predictions?*
- *How are the bridging algorithms derived and can they be derived using a different approach?*

To answer these questions, the following tasks are proposed:

- Study the theory about Bayesian probability distribution, and how this is used to derive a Bayesian filter and smoother, to provide a good understanding of the subject.
- Derive the bridging equations in detail using the partial derivation given in the articles as guidelines.

- Evaluate the performance of the algorithms in intent inference using real data measurements. Furthermore, using the same framework, evaluate its performance in predicting future states.

1.3 Outline of thesis

The thesis is divided into 8 chapter. Chapter 2 provides background material, such as theorems and the derivation of Bayesian filtering and smoothing. In chapter 3 the Bayesian bridging distribution is described and the different bridging methods are derived. Chapter 4 describes the motion models used for the study. Chapter 5 describes the test scheme and the different scenarios tested. The results are then provided in chapter 6. Chapter 7 provides the concluding remarks. Lastly, suggested future work is presented in chapter 8.

Chapter 2

Background material

This chapter is partly based on the work done in the preliminary specialisation project [7]. The chapter starts by providing the theorems and definitions necessary for the derivations in later sections. Thereafter, the general stochastic space model is introduced along with its discretization. The general Bayesian filtering and smoothing equations are derived, and Kalman filter (KF) and Rauch-Tung-Striebel (RTS) smoother are introduced. Lastly, a short introduction to modelling a marine vessel is presented. The majority of this chapter is based on Simo Särkkä's book *Bayesian Filtering and Smoothing* [8], and will be indicated if otherwise.

2.1 Theorems and definitions

Definition 1 (Gaussian distribution) *The Gaussian distribution of a random variable* $\mathbf{x} \in \mathbb{R}^n$ *with mean* $\mu \in \mathbb{R}^n$ *and covariance* $\mathbf{P} \in \mathbb{R}^{n \times n}$ *has a probability density on the form*

$$\mathcal{N}(\mathbf{x}; \mu, \mathbf{P}) = \frac{1}{(2\pi)^{n/2} |\mathbf{P}|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{P}^{-1}(\mathbf{x}-\mu)) \tag{2.1}$$

where $|\mathbf{P}|$ *is the determinant of* $\mathbf{P}$.

Lemma 1 (Woodbury matrix identity) *Let* $\mathbf{A} \in \mathbb{R}^{n \times n}$*,* $\mathbf{U} \in \mathbb{R}^{n \times m}$*,* $\mathbf{B} \in \mathbb{R}^{m \times m}$ *and* $\mathbf{V} \in \mathbb{R}^{m \times n}$ *be matrices, then the inverse of the sum* $\mathbf{A} + \mathbf{UBV}$ *will have the form*

$$(\mathbf{A} + \mathbf{UBV})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{B}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{A}^{-1} \tag{2.2}$$

Lemma 2 (Bayes' rule) *The probability of A happening given that B has happened is given by the following relation*

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{2.3}$$

With three events the equation becomes

$$p(A|B,C) = \frac{p(B|A,C)p(A|C)}{p(B|C)} \tag{2.4}$$

Lemma 3 (Chain rule for probability distributions) *A joint distribution of events can be represented as only conditional probability distributions.*

$$p(A,B,C) = p(A|B,C)p(B,C) = p(A|B,C)p(B|C)p(C) \tag{2.5}$$

Lemma 4 (Conditional distribution of Gaussian variables) *If the random variables* $\mathbf{x}$ *and* $\mathbf{y}$ *have the joint Gaussian probability distribution*

$$p(\mathbf{x},\mathbf{y}) = \mathscr{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix};\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix},\begin{bmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^T & \mathbf{B}\end{bmatrix}\right) \tag{2.6}$$

then their marginal and conditional distributions are given as

$$\begin{aligned} p(\mathbf{x}) &= \mathscr{N}(\mathbf{x};\mathbf{a},\mathbf{A})\\ p(\mathbf{y}) &= \mathscr{N}(\mathbf{y};\mathbf{b},\mathbf{B})\\ p(\mathbf{x}|\mathbf{y}) &= \mathscr{N}\left(\mathbf{x};\mathbf{a}+\mathbf{C}\mathbf{B}^{-1}(\mathbf{y}-\mathbf{b}),\ \mathbf{A}-\mathbf{C}\mathbf{B}^{-1}\mathbf{C}^T\right)\\ p(\mathbf{y}|\mathbf{x}) &= \mathscr{N}\left(\mathbf{y};\mathbf{b}+\mathbf{C}^T\mathbf{A}^{-1}(\mathbf{x}-\mathbf{a}),\ \mathbf{B}-\mathbf{C}^T\mathbf{A}^{-1}\mathbf{C}\right) \end{aligned} \tag{2.7}$$

Theorem 1 (Gaussian product identity I) *Let a random variable* $\mathbf{x} \in \mathbb{R}^n$ *have mean* $\mu \in \mathbb{R}^n$ *and covariance* $\mathbf{P} \in \mathbb{R}^{n\times n}$*, and another random variable* $\mathbf{y} \in \mathbb{R}^m$ *have mean* $\mathbf{Hx} \in \mathbb{R}^m$ *and covariance* $\mathbf{R} \in \mathbb{R}^{n\times n}$*, where* $\mathbf{H} \in \mathbb{R}^{m\times n}$ *and* $m \leq n$*. Then the product of the two Gaussian distributions results in the joint distribution*

$$\mathscr{N}(\mathbf{x};\mu,\mathbf{P})\mathscr{N}(\mathbf{y};\mathbf{Hx},\mathbf{R}) = \mathscr{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix};\begin{bmatrix}\mu\\ \mathbf{H}\mu\end{bmatrix},\begin{bmatrix}\mathbf{P} & \mathbf{P}\mathbf{H}^T\\ \mathbf{HP} & \mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R}\end{bmatrix}\right). \tag{2.8}$$

Using lemmas 3 and 4, the joint distribution can be written as

$$\begin{aligned} p(\mathbf{x},\mathbf{y}) &= p(\mathbf{x}|\mathbf{y})p(\mathbf{y})\\ &= \mathscr{N}(\mathbf{x};\mathbf{c},\mathbf{C})\mathscr{N}(\mathbf{y}\ ;\ \mathbf{H}\mu,\ \mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R}) \end{aligned} \tag{2.9}$$

where

$$\begin{aligned} \mathbf{C} =&\ \mathbf{P}-\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{HP}\\ \mathbf{c} =&\ \mu+\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R})^{-1}(\mathbf{y}-\mathbf{H}\mu) \end{aligned}$$

Theorem 2 (Gaussian product identity II) [9] *Let a random variable* $\mathbf{x} \in \mathbb{R}^n$ *have mean* $\mu \in \mathbb{R}^n$ *and covariance* $\mathbf{P} \in \mathbb{R}^{n\times n}$*, and another random variable* $\mathbf{y} \in \mathbb{R}^m$ *have mean* $\mathbf{Hx} \in \mathbb{R}^m$ *and covariance* $\mathbf{R} \in \mathbb{R}^{n\times n}$*, where* $\mathbf{H} \in \mathbb{R}^{m\times n}$ *and* $m \leq n$*. The product of the two Gaussian distributions will be on the form*

$$\mathscr{N}(\mathbf{x};\mu,\mathbf{P})\mathscr{N}(\mathbf{y};\mathbf{Hx},\mathbf{R}) = \mathscr{N}(\mathbf{x};\mathbf{c},\mathbf{C})\mathscr{N}(\mathbf{y}\ ;\ \mathbf{H}\mu,\ \mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R}) \tag{2.10}$$

where

$$\begin{aligned} \mathbf{C} =&\ (\mathbf{P}^{-1}+\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\\ \mathbf{c} =&\ \mathbf{C}(\mathbf{P}^{-1}\mu+\mathbf{H}^T\mathbf{R}^{-1}\mathbf{y}) \end{aligned}$$

2.2 Stochastic state space models

The stochastic state space model in continuous time is given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{G}\mathbf{n} \tag{2.11a}$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{w} \tag{2.11b}$$

where, $\mathbf{x}$ is the state vector, $\mathbf{y}$ is the measurement vector, $\mathbf{u}$ is the control input, and $\mathbf{n}$ and $\mathbf{w}$ are the process noise and the measurement noise, respectively. These are assumed to be Gaussian white noise with zero mean, and can be described by eq. (2.12).

$$\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{q}\delta(t-\tau)) \tag{2.12a}$$

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{r}\delta(t-\tau)) \tag{2.12b}$$

Here, $\mathbf{q}$ and $\mathbf{r}$ are the covariance matrices of the noises. For a system shown in eq. (2.11) that is linear time invariant (LTI), the exact solution is given by

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{G}\mathbf{n}(\tau)d\tau \tag{2.13a}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{w}(t) \tag{2.13b}$$

For practical implementations, e.g. for Kalman filtering, the system is discretized by integrating over the time intervals. Here a notation of t_k is used to represent time at iteration k, and $\mathbf{x}_k = \mathbf{x}(t_k)$ for simplicity. The discretization becomes

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{u}_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}\delta(t-\tau)) \tag{2.14a}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}\delta(t-\tau)) \tag{2.14b}$$

where

$$\mathbf{F} = e^{\mathbf{A}(t_k - t_{k-1})}, \tag{2.15a}$$

$$\mathbf{u}_k = \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{B}\mathbf{u}(\tau)d\tau, \tag{2.15b}$$

$$\mathbf{v}_k = \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{G}\mathbf{n}(\tau)d\tau, \tag{2.15c}$$

$$\mathbf{Q} = E[\mathbf{v}_k\mathbf{v}_k^T] = \int_{t_{k-1}}^{t_k} e^{\mathbf{A}(t_k-\tau)}\mathbf{G}\mathbf{q}\mathbf{G}^T e^{\mathbf{A}^T(t_k-\tau)}d\tau \tag{2.15d}$$

2.3 Filtering and Smoothing

2.3.1 Bayesian filtering

Let a probabilistic state space model be on the form shown in eq. (2.16) with its (a) dynamic model and (b) measurement model.

$$\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{2.16a}$$

$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}_k) \tag{2.16b}$$

These are assumed Markovian, i.e. "$\mathbf{x}_k$ *given* $\mathbf{x}_{k-1}$ *is independent of anything that has happened before the time step* $k-1$..." and "*...the current measurement* $\mathbf{y}_k$ *given the current state* $\mathbf{x}_k$ *is conditionally independent of the measurement and state histories*" [8]. This is illustrated in eq. (2.17), respectively. The notation $\mathbf{x}_{1:k}$ represents all states between, and including, the time steps 1 and k. i.e. $\mathbf{x}_{1:k} = \{\mathbf{x}_1, ..., \mathbf{x}_k\}$.

$$p(\mathbf{x}_k|\mathbf{x}_{1:k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{2.17a}$$

$$p(\mathbf{y}_k|\mathbf{x}_{1:k}, \mathbf{y}_{1:k-1}) = p(\mathbf{y}_k|\mathbf{x}_k) \tag{2.17b}$$

Furthermore, eq. (2.18) describes (a) the joint prior distribution of the states $p(\mathbf{x}_{0:L})$, and (b) the joint likelihood of the measurements $p(\mathbf{y}_{1:L}|\mathbf{x}_{0:L})$ for L time steps.

$$p(\mathbf{x}_{0:L}) = p(\mathbf{x}_0)p(\mathbf{x}_1|\mathbf{x}_0)...p(\mathbf{x}_L|\mathbf{x}_{L-1}) = p(\mathbf{x}_0)\prod_{k=1}^{L} p(\mathbf{x}_k|\mathbf{x}_{k-1}) \tag{2.18a}$$

$$p(\mathbf{y}_{1:L}|\mathbf{x}_{0:L}) = p(\mathbf{y}_1|\mathbf{x}_1)p(\mathbf{y}_2|\mathbf{x}_2)...p(\mathbf{y}_L|\mathbf{x}_L) = \prod_{k=1}^{L} p(\mathbf{y}_k|\mathbf{x}_k) \tag{2.18b}$$

Now, using Bayes' rule the posterior distribution of the measurement-conditioned-states is given in eq. (2.19). However, this will become very computational heavy to calculate for each new measurement as L gets large, and is therefore impractical to do in real time applications.

$$p(\mathbf{x}_{0:L}|\mathbf{y}_{1:L}) = \frac{p(\mathbf{y}_{1:L}|\mathbf{x}_{0:L})p(\mathbf{x}_{0:L})}{p(\mathbf{y}_{1:L})} \propto p(\mathbf{y}_{1:L}|\mathbf{x}_{0:L})p(\mathbf{x}_{0:L}) \tag{2.19}$$

The joint distribution of $\mathbf{x}_k$ and $\mathbf{x}_{k-1}$ given the measurement $\mathbf{y}_{k-1}$ is given in eq. (2.20). Integrating eq. (2.20) over $\mathbf{x}_{k-1}$ results in the prediction step of the Bayes filter given in eq. (2.21a) (also called the Chapman-Kolmogorov equation). The update step is given by the Bayes rule given in eq. (2.21b), where the term $p(\mathbf{y}_k|\mathbf{y}_{1:k-1})$ is called the normalization constant, denoted as Z_k. The combination of the equations in 2.21 are called the Bayes filter.

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) = &\quad p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{y}_{1:k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \\ = &\quad p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \end{aligned} \tag{2.20}$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1} \tag{2.21a}$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \propto p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \tag{2.21b}$$

where the normalisation constant is given by

$$Z_k := p(\mathbf{y}_k|\mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{y}_k|\mathbf{y}_{1:k-1})d\mathbf{x}_k \tag{2.22}$$

2.3.2 Bayesian smoothing

The marginal posterior distribution of the state $\mathbf{x}_k$ can be computed with a smoother using the future measurements up to step $L > k$, i.e. the distribution of $\mathbf{x}_k$ given the measurements $\mathbf{y}_{1:L}$, where $L > k$:

$$p(\mathbf{x}_k|\mathbf{y}_{1:L}) \tag{2.23}$$

The smoothed distribution in eq. (2.23) can be found for any $k < L$ by going backwards in time. At the last time step $k = L$ both the filtering and the smoothing distribution coincide, $p(\mathbf{x}_L|\mathbf{y}_{1:L})$. The same assumption are made as in section 2.3.1, that the model is Markovian, i.e. $p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:L}) = p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:k})$. With Bayes's rule this can be written as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:L}) &= p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:k}) \\ &= \frac{p(\mathbf{x}_k,\mathbf{x}_{k+1}|\mathbf{y}_{1:k})}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})} \\ &= \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k,\mathbf{y}_{1:k})p(\mathbf{x}_k|\mathbf{y}_{1:k})}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})} \\ &= \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})} \end{aligned} \tag{2.24}$$

Now, assuming the smoothing distribution of the next step, $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:L})$, is available, the joint distribution of $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ given $\mathbf{y}_{1:L}$ can be written as

$$\begin{aligned} p(\mathbf{x}_k,\mathbf{x}_{k+1}|\mathbf{y}_{1:L}) &= p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:L})p(\mathbf{x}_{k+1}|\mathbf{y}_{1:L}) \\ &= p(\mathbf{x}_k|\mathbf{x}_{k+1},\mathbf{y}_{1:k})p(\mathbf{x}_{k+1}|\mathbf{y}_{1:L}) \\ &= \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})p(\mathbf{x}_{k+1}|\mathbf{y}_{1:L})}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})} \end{aligned} \tag{2.25}$$

The marginal distribution of $\mathbf{x}_k$ and $\mathbf{y}_{1:L}$ is found by integrating eq. (2.25) over $\mathbf{x}_{k+1}$, which is the backward update step of the Bayesian smoother, and is given in eq. (2.26b). The prediction step of the Bayesian smoother is the same as the prediction step of the Bayesian filter given in eq. (2.21a), it is again shown in eq. (2.26a) for another step. Together, eq. (2.26) forms the smoothing equations.

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k}) = \int p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k})d\mathbf{x}_k \tag{2.26a}$$

$$p(\mathbf{x}_k|\mathbf{y}_{1:L}) = p(\mathbf{x}_k|\mathbf{y}_{1:k})\int \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k)p(\mathbf{x}_{k+1}|\mathbf{y}_{1:L})}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})}d\mathbf{x}_{k+1} \tag{2.26b}$$

2.3.3 Linear filtering and smoothing

Consider the following discrete Gaussian and LTI system given in eqs. (2.27a) and (2.27b), where the initial state is distributed according to eq. (2.27c).

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0},\mathbf{Q}) \tag{2.27a}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0},\mathbf{R}) \tag{2.27b}$$

$$\mathbf{x}_1 \sim \mathcal{N}(\mu_1,\mathbf{P}_1) \tag{2.27c}$$

Furthermore, the distributions for the predicted state and current time measurement are given by eqs. (2.28a) and (2.28b), respectively.

$$p(\mathbf{x}_k|\mathbf{x}_{k-1}) = \mathcal{N}(\mathbf{x}_k;\mathbf{F}\mathbf{x}_{k-1},\mathbf{Q}) \tag{2.28a}$$

$$p(\mathbf{y}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{y}_k;\mathbf{H}\mathbf{x}_k,\mathbf{R}) \tag{2.28b}$$

The joint distribution of the state **x** and the measurement **y** is given as

$$p(\mathbf{x},\mathbf{y}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix};\begin{bmatrix}\mu\\ \mathbf{H}\mu\end{bmatrix},\begin{bmatrix}\mathbf{P} & \mathbf{P}\mathbf{H}^T\\ \mathbf{H}\mathbf{P} & \mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R}\end{bmatrix}\right) \tag{2.29}$$

Then their marginal distributions can be written as

$$\begin{aligned} p(\mathbf{x}) &= \mathcal{N}(\mu,\mathbf{P}) \\ p(\mathbf{y}) &= \mathcal{N}(\mathbf{H}\mu,\mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R}) \end{aligned} \tag{2.30}$$

and their conditional distributions is given as

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &= \mathcal{N}\left(\mathbf{x};\ \mu+\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R})^{-1}(\mathbf{y}-\mathbf{H}\mu),\ \mathbf{P}-\mathbf{P}\mathbf{H}^T(\mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R})^{-1}\mathbf{H}\mathbf{P}\right) \\ p(\mathbf{y}|\mathbf{x}) &= \mathcal{N}\left(\mathbf{y};\ \mathbf{H}\mu+\mathbf{H}\mathbf{P}\mathbf{P}^{-1}(\mathbf{x}-\mu),\ (\mathbf{H}\mathbf{P}\mathbf{H}^T+\mathbf{R})-\mathbf{H}\mathbf{P}\mathbf{P}^{-1}\mathbf{P}\mathbf{H}^T\right) \\ &= \mathcal{N}\left(\mathbf{y};\ \mathbf{H}\mu+\mathbf{H}(\mathbf{x}-\mu),\ \mathbf{R}\right) \end{aligned} \tag{2.31}$$

From the above information the Kalman filter (KF) can be derived, which is the solution to the Bayesian filtering equations in eq. (2.21) for a linear Gaussian system. The KF algorithm is shown in algorithm 1 where the hat-notation above a state, or a measurement, illustrates its estimate. The reader is referred to [8, pp.56-58] for more detailed information about the derivation of the KF.

Algorithm 1 Kalman filter (KF)

```
procedure KF($\hat{\mathbf{x}}_{k-1},\mathbf{P}_{k-1},\mathbf{y}_k$)
    $\hat{\mathbf{x}}_{k|k-1} \leftarrow \mathbf{F}\hat{\mathbf{x}}_{k-1}$                        ▷ Predicted state estimate
    $\mathbf{P}_{k|k-1} \leftarrow \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T+\mathbf{Q}$                 ▷ Predicted covariance
    $\hat{\mathbf{y}}_{k|k-1} \leftarrow \mathbf{H}\hat{\mathbf{x}}_{k|k-1}$                      ▷ Predicted measurement
    $\upsilon_k \leftarrow \mathbf{y}_k-\hat{\mathbf{y}}_{k|k-1}$                          ▷ Innovation measurement
    $\mathbf{J}_k \leftarrow \mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T+\mathbf{R}$                  ▷ Innovation covariance
    $\mathbf{W}_k \leftarrow \mathbf{P}_{k|k-1}\mathbf{H}^T\mathbf{J}_k^{-1}$                      ▷ Kalman gain
    $\hat{\mathbf{x}}_k \leftarrow \hat{\mathbf{x}}_{k|k-1}+\mathbf{W}_k\nu_k$                     ▷ Posterior state estimate
    $\mathbf{P}_k \leftarrow (\mathbf{I}-\mathbf{W}_k\mathbf{H})\mathbf{P}_{k|k-1}$                ▷ Posterior covariance
    return $\hat{\mathbf{x}}_k,\ \mathbf{P}_k,\ \hat{\mathbf{x}}_{k|k-1},\ \mathbf{P}_{k|k-1},\ \mathbf{J}_k$
end procedure
```

From the KF the posterior state distribution conditioned on the measurements is given by

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \mathcal{N}(\mathbf{x}_k;\hat{\mathbf{x}}_k,\mathbf{P}_k) = \mathcal{N}(\mathbf{x}_k;\mu_k,\mathbf{P}_k) \tag{2.32}$$

With the assumption that the time step interval is fixed the Rauch-Tung-Striebel (RTS) smoother can be derived given eqs. (2.27)–(2.31). Firstly, using Gaussian distribution computational rules, the joint distribution of the state $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ given the measurement $\mathbf{y}_{1:k}$ can be expressed as

$$\begin{aligned} p(\mathbf{x}_k, \mathbf{x}_{k+1} | \mathbf{y}_{1:k}) &= p(\mathbf{x}_{k+1}|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{y}_{1:k}) \\ &= \mathcal{N}(\mathbf{x}_{k+1}; \mathbf{F}\mathbf{x}_k, \mathbf{Q}) \mathcal{N}(\mathbf{x}_k; \mu_k, \mathbf{P}_k) \\ &= \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}; \begin{bmatrix} \mu_k \\ \mathbf{F}\mu_k \end{bmatrix}, \begin{bmatrix} \mathbf{P}_k & \mathbf{P}_k\mathbf{F}^T \\ \mathbf{F}\mathbf{P}_k & \mathbf{F}\mathbf{P}_k\mathbf{F}^T + \mathbf{Q} \end{bmatrix} \right) \\ &:= \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \end{bmatrix}; \nu_1, \mathbf{p}_1 \right) \end{aligned} \tag{2.33}$$

Secondly, using the conditioning rule, the joint distribution of $\mathbf{x}_k$ given $\mathbf{x}_{k+1}$ and all measurements $\mathbf{y}_{1:L}$ can be expressed as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{y}_{1:L}) &= p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{y}_{1:k}) \\ &= \mathcal{N}\left(\mathbf{x}_k;\ \mu_k + \mathbf{G}_k(\mathbf{x}_{k+1} - \mathbf{F}\mu_k),\ \mathbf{P}_k - \mathbf{G}_k(\mathbf{F}\mathbf{P}_k\mathbf{F}^T + \mathbf{Q})\mathbf{G}_k^T\right) \\ &:= \mathcal{N}(\mathbf{x}_k; \nu_2, \mathbf{p}_2) \end{aligned} \tag{2.34}$$

where

$$\mathbf{G}_k = \mathbf{P}_k\mathbf{F}^T(\mathbf{F}\mathbf{P}_k\mathbf{F}^T + \mathbf{Q})^{-1} \tag{2.35}$$

Finally, the joint distribution of $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$ given all measurements $\mathbf{y}_{1:L}$ can be expressed as

$$\begin{aligned} p(\mathbf{x}_{k+1}, \mathbf{x}_k|\mathbf{y}_{1:L}) &= p(\mathbf{x}_k|\mathbf{x}_{k+1}, \mathbf{y}_{1:L}) p(\mathbf{x}_{k+1}|\mathbf{y}_{1:L}) \\ &= \mathcal{N}(\mathbf{x}_k; \nu_2, \mathbf{p}_2) \mathcal{N}(\mathbf{x}_{k+1}; \mu^s_{k+1}, \mathbf{P}^s_{k+1}) \\ &= \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}_k \end{bmatrix}; \begin{bmatrix} \mu^s_{k+1} \\ \mu_k + \mathbf{G}_k(\mu^s_{k+1} - \mathbf{F}\mu_k) \end{bmatrix}, \begin{bmatrix} \mathbf{P}^s_{k+1} & \mathbf{P}^s_{k+1}\mathbf{G}_k^T \\ \mathbf{G}_k\mathbf{P}^s_{k+1} & \mathbf{G}_k\mathbf{P}^s_{k+1}\mathbf{G}_k^T + \mathbf{p}_2 \end{bmatrix} \right) \\ &:= \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{x}_k \end{bmatrix}; \nu_3, \mathbf{p}_3 \right) \end{aligned} \tag{2.36}$$

Here the s-superscript is used to refer to smoothing, e.g. μ^s_k is the smoothing mean at time step k. With the marginal mean and covariance from eq. (2.36) the smoothing distribution is Gaussian and can be written as

$$p(\mathbf{x}_{k+1}|\mathbf{y}_{1:L}) = \mathcal{N}(\mathbf{x}_k; \mu^s_k, \mathbf{P}^s_k) \tag{2.37}$$

where

$$\begin{aligned} \mu^s_k &= \mu_k + \mathbf{G}_k(\mu^s_{k+1} - \mathbf{F}\mu_k) \\ \mathbf{P}^s_k &= \mathbf{P}_k + \mathbf{G}_k(\mathbf{P}^s_{k+1} - \mathbf{F}\mathbf{P}_k\mathbf{F}^T - \mathbf{Q})\mathbf{G}_k^T \end{aligned}$$

Here the mean μ_k and the covariance $\mathbf{P}_k$ are calculated with KF. The backward recursion begins at the last time step $k = L$ where $\mu^s_k = \mu_k$ and $\mathbf{P}^s_k = \mathbf{P}_k$.

Algorithm 2 Rauch-Tung-Striebel (RTS) smoother

```
procedure RTS($\mu_k$, $\mathbf{P}_k$, $\hat{\mu}_{k+1}$, $\mathbf{P}^s_{k+1}$)
    Compute mean $\mu_k$ and covariance $\mathbf{P}_k$ with KF.
    $\hat{\mu}_{k+1} \leftarrow \mathbf{F}\mu_k$                                    ▷ Predict next step mean
    $\hat{\mathbf{P}}_{k+1} \leftarrow \mathbf{F}\mathbf{P}_k\mathbf{F}^T + \mathbf{Q}$                          ▷ Predict next step covariance
    $\mathbf{G}_k \leftarrow \mathbf{P}_k\mathbf{F}^T\hat{\mathbf{P}}^{-1}_{k+1}$
    $\mu^s_k \leftarrow \mu_k + \mathbf{G}_k(\mu^s_{k+1} - \hat{\mu}_{k+1})$                  ▷ Previous smoothing mean
    $\mathbf{P}^s_k \leftarrow \mathbf{P}_k + \mathbf{G}_k(\mathbf{P}^s_{k+1} - \hat{\mathbf{P}}_{k+1})\mathbf{G}^T_k$          ▷ Previous smoothing covariance
    return $\mu^s_k$, $\mathbf{P}^s_k$
end procedure
```

2.4 Marine vessels in motion

Thor I. Fossen [10] presents a detailed description of the mathematical models for marine vessels in his *Handbook of Marine Craft Hydrodynamics and Motion Control* [10]. From which, a short introduction to modelling of marine vessels is presented, to provide a basic understanding of their behaviour.

A marine vessel that can move freely in 3D space is represented by a maximum of 6 degrees of freedom (DoF). These represent position along the x, y and z-axes, and the orientation around said axes, represented by Euler angles. The state vector is defined as $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^T$. Furthermore, the time derivatives represent the translation in x, y and z direction and its rotation about their axes, called *surge, sway, heave, roll, pitch* and *yaw*, respectively, with the notation $\nu = [u, v, w, p, q, r]^T$. This is illustrated in fig. 2.1 for the body {b}-frame of a marine vessel, $\{b\}$ = $[x_b, y_b, z_b]$. Table 2.1 lists the notations for its position, angles, and its corresponding velocities and forces/moments.

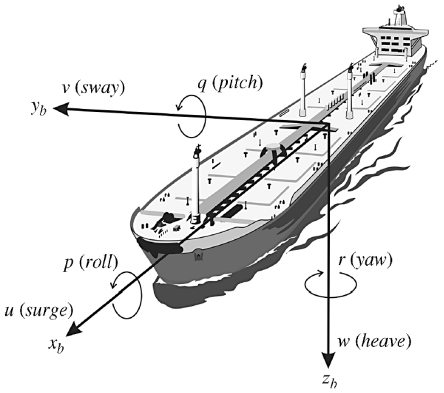


Figure 2.1: The 6 DoF velocities of a marine vessel in its $\{b\}$-frame [10, pp.16].

Table 2.1: Notation for marine vessels according to The Society of Naval Architects and Marine Engineers (SNAME) [11]

DoF description	Forces and moments	Linear and angular velocities	Position and Euler angles
Motion in the x direction, *surge*	X	u	x
Motion in the y direction, *sway*	Y	v	y
Motion in the z direction, *heave*	Z	w	z
Rotation about the x axis, *roll*	K	p	ϕ
Rotation about the y axis, *pitch*	M	q	θ
Rotation about the z axis, *yaw*	N	r	ψ

A common reference frame is the North-East-Down (NED) coordinate system $\{n\} = [x_n, y_n, z_n]$, where x-axis points towards true north, y-axis towards east, and z-axis down towards the centre of the earth.

For a surface vessels the 6 DoF model can be simplified to a 3 DoF model, where *heave*, *roll*, and *pitch* are all assumed zero. Therefore, the corresponding states can be ignored, decreasing the state vector to $\mathbf{x} = [x, y, \psi]^T$ and the velocity vector to $\nu = [u, v, r]^T$. This will produce relative accurate model for surface vessels as long as there is not too much wind or waves affecting the vessel [10].

Chapter 3

Destination prediction using bridging distribution

Ahmad et al. [5] and Liang et al. [6] introduced models for predicting the intended destination of a tracked object. This chapter presents these models in detail along with their mathematical derivations.

3.1 Systems general tracking model

The discretization of a linear-Gaussian tracking model used for destination inference, with time step $T = t_k - t_{k-1}$, can be written as

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{m} + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathscr{N}(\mathbf{0}, \mathbf{Q}\delta(t-\tau)) \tag{3.1a}$$

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathscr{N}(\mathbf{0}, \mathbf{R}\delta(t-\tau)) \tag{3.1b}$$

where $\mathbf{F}$, $\mathbf{m}$ and $\mathbf{Q}$ are all functions of the time step T and a given destination d, i.e. $\mathbf{F} = \mathbf{F}(T,d)$. $\mathbf{v}$ and $\mathbf{w}$ are the process and measurement noises, respectively, which are both assumed Gaussian and zero mean. Equation (3.1a) differ slightly from the conventional tracking model, where the vector $\mathbf{m}$ is added to the equation. Depending on the specifics of the motion model, this vector will help the state revert towards desired values, or for some models it becomes zero, e.g. the constant velocity (CV) model.

Assume there is a set of N possible destinations $\mathscr{D} = \{d = d_1, ..., d_N\}$ which includes all possible destinations the tracked object can be going towards. The destination $d_n \in \mathscr{D}$, $\forall n = 1, ..., N$ is modelled as a Gaussian a priori distribution, i.e. $d_n \sim \mathscr{N}(\mathbf{x}_d, \Sigma_d)$, rather than a single point. In two DoF (xy-coordinates) the destination mean is given by $\mathbf{x}_d = [x_d, y_d, 0, 0]^T$, where $\mathbf{x} = [x, y, \dot{x}, \dot{y}]^T$, and the object will reach its destination at a time step f, at time t_f. The final state $\mathbf{x}_f$ is therefore assumed to be a random variable with the Gaussian distribution $\mathscr{N}(\mathbf{x}_f; \mathbf{x}_d, \Sigma_d)$. This can be viewed as an ellipse with its centre at $\mathbf{x}_d$. The tracked object is assumed to arrive at its destination on the time interval $t_f = [t_{fmin}, t_{fmax}]$, assuming t_{fmin} is the least amount of time it takes for the object to travel towards its destination, and t_{fmax} the most amount of time. For a discrete arrival time, the interval is evenly divided between q quadrature points. In other words, there exists a set $\mathscr{T} = \{t_f = t_{f1}, ...t_{fq}\}$ that includes all possible arrival times, where $t_{f1} < t_{f2}... < t_{fq}$. Abbreviations for the transition functions in eq. (3.1), with different time steps, are listed in table 3.1. Using these abbreviations, the system's structure is illustrated in fig. 3.1.

Table 3.1: Abbreviations for the transition functions in the system in fig. 3.1 for different time steps.

t	$\mathbf{F}(t,d_n)$	$\mathbf{m}(t,d_n)$	$\mathbf{Q}(t,d_n)$
$t_k - t_{k-1}$	$\mathbf{F}_T$	$\mathbf{m}_T$	$\mathbf{Q}_T$
$t_f - t_k$	$\mathbf{F}_f$	$\mathbf{m}_f$	$\mathbf{Q}_f$
$t_f - t_{k-1}$	$\mathbf{F}_{f-1}$	$\mathbf{m}_{f-1}$	$\mathbf{Q}_{f-1}$

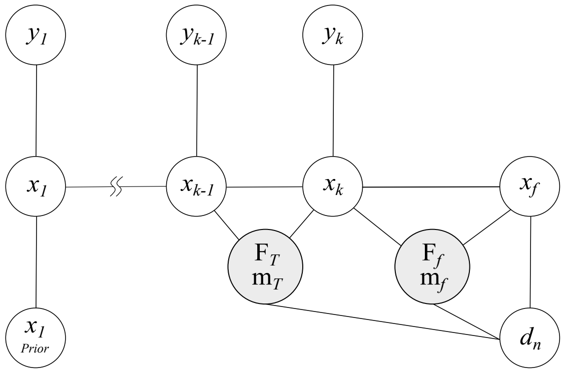


Figure 3.1: The structure of the system after k measurements. The transition matrix and vector are a function of the time step between states and the destination d_n. [5]

3.2 Bayesian distribution

3.2.1 Arrival time distribution

To derive the arrival time distribution the destination, d_n, is assumed known. Using Bayes' rule, the conditioned arrival time distribution is given by

$$\begin{aligned} p(t_f|d_n,\mathbf{y}_{1:k}) &\propto \frac{p(\mathbf{y}_{1:k}|t_f,d_n)p(t_f|d_n)}{p(\mathbf{y}_{1:k}|d_n)} && (3.2a) \\ &= \frac{p(\mathbf{y}_{1:k}|t_f,d_n)p(t_f|d_n)}{\int p(\mathbf{y}_{1:k}|t_f,d_n)p(t_f|d_n)dt_f} && (3.2b) \\ &\propto p(\mathbf{y}_{1:k}|t_f,d_n)p(t_f|d_n) && (3.2c) \end{aligned}$$

where $p(\mathbf{y}_{1:k}|d_n)$ is the normalisation constant, and $p(t_f|d_n)$ is the prior distribution of the possible arrival times $t_{fi}, \forall i = 1, ...q$, which for simplicity, could be assumed uniformly distributed between t_{f1} and t_{fq}, i.e. $p(t_f|d_n) = \mathscr{U}(t_{f1}, t_{fq})$. The measurement conditioned distribution $p(\mathbf{y}_{1:k}|t_f,d_n)$ for a given arrival time t_{fi} and a destination d_n is given in eq. (3.3).

$$\begin{aligned} p(\mathbf{y}_{1:k}|t_{fi},d_n) &= p(\mathbf{y}_{1:k-1}|t_{fi},d_n)p(\mathbf{y}_k|\mathbf{y}_{1:k-1},t_{fi},d_n) && (3.3a) \\ &= p(\mathbf{y}_1|t_{fi},d_n)\prod_{m=2}^{k} p(\mathbf{y}_m|\mathbf{y}_{1:m-1},t_{fi},d_n) && (3.3b) \end{aligned}$$

For a discretization of the arrival time interval the integral becomes a sum over the quadrature points. This leads to the weighted likelihood of a given arrival time and a destination given by

$$p(t_{fi}|d_n,\mathbf{y}_{1:k}) \approx \frac{p(\mathbf{y}_{1:k}|t_{fi},d_n)p(t_{fi}|d_n)}{\sum_{i=1}^{q} p(\mathbf{y}_{1:k}|t_{fi},d_n)p(t_{fi}|d_n)} \tag{3.4}$$

and the approximated arrival time distribution is then given by

$$p(t_f|d_n,\mathbf{y}_{1:k}) \approx \sum_{i=1}^{q} p(t_{fi}|d_n,\mathbf{y}_{1:k})\delta_{\{t_{fi}\}} \tag{3.5}$$

where $\delta_{\{t_{fi}\}}$ is the Direc delta function at t_{fi}.

3.2.2 Destination inference

The measurement likelihood, for an unknown arrival time, can be found by integrating the arrival time conditioned likelihood in eq. (3.2c) over the arrival time interval, which gives

$$p(\mathbf{y}_{1:k}|d_n) = \int_{t_{f1}}^{t_{fq}} p(\mathbf{y}_{1:k}|t_f,d_n)p(t_f|d_n)dt_f \tag{3.6}$$

where $p(t_f|d_n)$ is the arrival time a priori for destination d_n. Since time is one dimensional, the integral in eq. (3.6) can be approximated using numerical quadrature, e.g. Simpson's rule quadrature scheme or the trapezoidal rule. The approximation of the integral over $t_f = [t_{f1},t_{fq}]$ using Simpson's rule, is shown in eq. (3.7), with an odd number q of evenly spaced quadrature points. Whereas, the trapezoidal rule is shown in eq. (3.8), where Δt_f is the time between the evenly spaced quadrature points.

$$\begin{aligned} p(\mathbf{y}_{1:k}|d_n) \approx \frac{t_{fq}-t_{f1}}{3(q-1)} \Big[\, & p(\mathbf{y}_{1:k}|t_{f1},d_n)p(t_{f1}|d_n) + p(\mathbf{y}_{1:k}|t_{fq},d_n)p(t_{fq}|d_n) \\ & + 4 \sum_{i=1}^{(q-1)/2} p(\mathbf{y}_{1:k}|t_{f(2i)},d_n)p(t_{f(2i)}|d_n) \\ & + 2 \sum_{i=1}^{(q-1)/2-1} p(\mathbf{y}_{1:k}|t_{f(2i+1)},d_n)p(t_{f(2i+1)}|d_n) \,\Big] \end{aligned} \tag{3.7}$$

$$\begin{aligned} p(\mathbf{y}_{1:k}|d_n) \approx \Delta t_f \Big[\, & \frac{1}{2}p(\mathbf{y}_{1:k}|t_{f1},d_n)p(t_{f1}|d_n) \; + \; \frac{1}{2}p(\mathbf{y}_{1:k}|t_{fq},d_n)p(t_{fq}|d_n) \\ & + \sum_{i=2}^{q-1} p(\mathbf{y}_{1:k}|t_{fi},d_n)p(t_{fi}|d_n) \,\Big] \end{aligned} \tag{3.8}$$

The *measurements-conditioned-destination* distribution is found by using Bayes' rule and is given as

$$p(d|\mathbf{y}_{1:k}) \; = \; \frac{p(\mathbf{y}_{1:k}|d)p(d)}{p(\mathbf{y}_{1:k})} \; \propto \; p(\mathbf{y}_{1:k}|d)p(d) \tag{3.9}$$

The weight on a certain destination $d = d_n$ can then be expressed as its probability divided by the sum of the probabilities of all destinations. A summation is used instead of an integration since the destinations are discontinuous. This results in the distribution

$$p(d_n|\mathbf{y}_{1:k}) \; = \; \frac{p(\mathbf{y}_{1:k}|d_n)p(d_n)}{\sum_{d\in\mathscr{D}} p(\mathbf{y}_{1:k}|d)p(d)} \tag{3.10}$$

where $p(d)$ is the prior distribution of all destinations $d \in \mathscr{D}$, which can be determined through historical data, or by choice, e.g. letting all N destinations have the same probability, $Pr(d) = 1/N \;, \forall d \in \mathscr{D}$.

To prevent underflow during computations, due to low valued likelihoods, one must use a logarithmic representation of these equations. Examples of this is given in Appendix A.I.

3.2.3 Arrival time distribution at any destination

The posterior arrival time distribution at any destination $d \in \mathscr{D}$ can be found by integrating over all destination points; or, in this case, summing over $d \in \mathscr{D}$ since these are discontinuous. This is shown in eq. (3.11) and can be approximated as weighted time points as shown in eq. (3.12)

$$\begin{aligned} p(t_f|\mathbf{y}_{1:k}) &= \sum_{d\in\mathscr{D}} p(t_f, d|\mathbf{y}_{1:k}) \\ &\propto \sum_{d\in\mathscr{D}} p(\mathbf{y}_{1:k}|t_f, d)p(t_f|d)p(d) \end{aligned} \tag{3.11}$$

$$p(t_f|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{q} \tilde{v}_i \delta_{\{t_{fi}\}} \tag{3.12}$$

with the weights defined as

$$\tilde{v}_i = \frac{\sum_{d\in\mathscr{D}} p(\mathbf{y}_{1:k}|t_{fi}, d)p(t_{fi}|d)p(d)}{\sum_{i=1}^{q}\sum_{d\in\mathscr{D}} p(\mathbf{y}_{1:k}|t_{fi}, d)p(t_{fi}|d)p(d)} \tag{3.13}$$

Note that it is assumed that the arrival time interval is the same for all destinations.

3.3 Bridging model using joint state

The concept of *bridging* the current state with the final state was first introduced by Ahmad et al. [5], where they used a joint state $\mathbf{z}_k$ to filter for $\mathbf{x}_k$. This joint state includes both the current state and the the final state, and is given by

$$\mathbf{z}_k = \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_f \end{bmatrix} \tag{3.14}$$

Assuming both the destination d_n and the arrival time t_f are known, the joint distribution, using Bayes' rule, is given by

$$p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{x}_f) \;=\; p(\mathbf{x}_f|\mathbf{x}_{k+1}, \mathbf{x}_k)p(\mathbf{x}_{k+1}|\mathbf{x}_k) \tag{3.15}$$

and the transition distribution of $\mathbf{z}_{k+1}$ conditioned on $\mathbf{z}_k$ is given by

$$\begin{aligned} p(\mathbf{z}_{k+1}|\mathbf{z}_k, d_n, t_f) &= p(\mathbf{x}_f, \mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{x}_f, d_n, t_f) && \text{(3.16a)} \\ &= p(\mathbf{x}_f|\mathbf{x}_{k+1}, \mathbf{x}_f, d_n, t_f)p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{x}_f, d_n, t_f) && \text{(3.16b)} \\ &\propto p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{x}_f, d_n, t_f) && \text{(3.16c)} \end{aligned}$$

This follows from that the two terms in eq. (3.16b) are independent of one another allowing the first term to be ignored. For a Gaussian distribution, eq. (3.16c) is given by

$$\begin{aligned} p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{x}_f, d_n, t_f) &\propto p(\mathbf{x}_f|\mathbf{x}_{k+1}, d_n, t_f)p(\mathbf{x}_{k+1}|\mathbf{x}_k, d_n, t_f) && \text{(3.17a)} \\ &= \mathscr{N}(\mathbf{x}_f; \mathbf{F}_f\mathbf{x}_{k+1} + \mathbf{m}_f, \mathbf{Q}_f)\mathscr{N}(\mathbf{x}_{k+1}; \mathbf{F}_T\mathbf{x}_k + \mathbf{m}_T, \mathbf{Q}_T) && \text{(3.17b)} \\ &\propto \mathscr{N}(\mathbf{x}_{k+1}; c_k, C_k) && \text{(3.17c)} \end{aligned}$$

with the following parameters

$$
\begin{aligned}
C_k &= \left(\mathbf{Q}_T^{-1} + \mathbf{F}_f^T \mathbf{Q}_f^{-1} \mathbf{F}_f\right)^{-1} \\
&= \mathbf{Q}_T - \mathbf{Q}_T \mathbf{F}_f^T \left(\mathbf{Q}_f + \mathbf{F}_f \mathbf{Q}_T \mathbf{F}_f^T\right) \mathbf{F}_f \mathbf{Q}_T
\end{aligned} \tag{3.18a}
$$

$$
\begin{aligned}
c_k &= C_k \left(\mathbf{Q}_T^{-1}(\mathbf{F}_T \mathbf{x}_k + \mathbf{m}_T) + \mathbf{F}_f^T \mathbf{Q}_f^{-1}(\mathbf{x}_f - \mathbf{m}_f)\right) \\
&= \left[C_k \mathbf{Q}_T^{-1} \mathbf{F}_T \; , \; C_k \mathbf{F}_f^T \mathbf{Q}_f^{-1}\right] \begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_f \end{bmatrix} + C_k(\mathbf{Q}_T^{-1} \mathbf{m}_T - \mathbf{F}_f^T \mathbf{Q}_f^{-1} \mathbf{m}_f)
\end{aligned} \tag{3.18b}
$$

$$
c_k := \mathbf{G}_k \mathbf{z}_k + \mathbf{b}_k \tag{3.18c}
$$

where $\mathbf{Q}_T$ and $\mathbf{Q}_f$ are the covariance matrices that are functions of the time step $t_k - t_{k-1}$ and the time step $t_f - t_k$, respectively. Similarly, $\mathbf{F}_T$ and $\mathbf{F}_f$ are the transition matrices that are functions of the time step $t_k - t_{k-1}$ and the time step $t_f - t_k$, respectively. Again, these parameters are provided in table 3.1 above. This allows the discretization of the joint state $\mathbf{z}_k$ to be written as

$$
\mathbf{z}_{k+1} = \mathbf{S}_k \mathbf{z}_k + \tilde{\mathbf{b}}_k + \gamma_k, \qquad \gamma_k \sim \mathcal{N}(\mathbf{0}, \mathbf{U}_k) \tag{3.19}
$$

with

$$
\mathbf{S}_k = \begin{bmatrix} \mathbf{G}_k \\ \mathbf{0}, \quad \mathbf{I} \end{bmatrix}, \qquad \tilde{\mathbf{b}}_k = \begin{bmatrix} \mathbf{b}_k \\ \mathbf{0} \end{bmatrix}, \qquad \mathbf{U}_k = \begin{bmatrix} \mathbf{C}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.20}
$$

The corresponding measurement model has the following form.

$$
\mathbf{y}_k = [\mathbf{H}, \; \mathbf{0}] \, \mathbf{z}_k + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{3.21}
$$

Since eqs. (3.19) and (3.21) make a linear Gaussian system, a KF can be applied to calculate the posterior estimates. For the first measurement, a prior for computing $\mathbf{z}_1 = [\mathbf{x}_1^T, \mathbf{x}_f^T]^T$ is required. The prior on $\mathbf{x}_1$ is the standard prior on the initial state, but the prior distribution on $\mathbf{x}_f$ is assumed to be $p(\mathbf{x}_f|d) = \mathcal{N}(\mathbf{x}_f; \mathbf{x}_d, \Sigma_d)$, and independent of the initial prior on $\mathbf{x}_1$. This allows the initial extended prior distribution conditioned on t_f to be written as

$$
p(\mathbf{z}_1 | t_f, d_n) = \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_f \end{bmatrix}; \begin{bmatrix} \mu_1 \\ \mathbf{x}_d \end{bmatrix}, \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \Sigma_d \end{bmatrix} \right) \tag{3.22}
$$

Ahmad et al. [5] presented an algorithm for destination inference and is shown in algorithm 3 below. It finds the posterior distribution over all $d \in \mathcal{D}$ after k measurements, $p(d|\mathbf{y}_{1:k})$. Then, the intended destination conditioned on the measurements is determined using maximum a posteriori (MAP) estimate:

$$
\hat{d}(t_n) = \underset{d=1,2,\ldots,N}{\arg\max} \; p(d|\mathbf{y}_{1:n}) \tag{3.23}
$$

3.3.1 State estimate and trajectory

Posterior state estimate is found by integrating over all destinations and arrival times, shown in eq. (3.24a). Furthermore, the distribution is Gaussian with mean $\hat{\mathbf{z}}_k$ and covariance $\mathbf{P}_k$ from algorithm 3, multiplied with $[\mathbf{I}, \mathbf{0}]$ to filter out $\mathbf{x}_k$ from $\mathbf{z}_k$. The integral can be approximated to

Algorithm 3 Destination inference [5]

```
procedure DI($\mathbf{y}_{1:n}$)
    Initialize Set initial likelihood $L_0^{(d,i)} = 1$ and $\hat{z}_0^{(d,i)}, \Sigma_0^{(d,i)}$ to the priors from eq. (3.22) for all $d \in \mathscr{D}$ and quadrature points $i$
    for measurements $k = 1, ..., n$ do
        for destination $d \in \mathscr{D}$ do
            for quadrature point $i \in 1, ..., q$ do
                Compute $\mathbf{S}_k^{(d,i)}$ and $\mathbf{U}_k^{(d,i)}$ in eq. (3.19) for measurement at time $t_k$, destination $d$ and arrival time $t_{fi}$.
                $\{\hat{\mathbf{z}}_k^{(d,i)}, \mathbf{P}_k^{(d,i)}, \hat{\mathbf{y}}_k, \mathbf{J}_k\} \leftarrow KF(\hat{\mathbf{x}}_{k-1}^{(d,i)}, \mathbf{P}_{k-1}^{(d,i)}, \mathbf{y}_{k|k-1})$    ▷ Run KF iteration (algorithm 1) to compute posterior state estimate and covariance, predicted measurement, and innovation covariance
                $\ell_k^{(d,i)} \leftarrow \mathcal{N}(\mathbf{y}_k; \hat{\mathbf{y}}_k, \mathbf{J}_k) = p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, d, t_{fi})$    ▷ Prediction error decomposition
                $L_k^{(d,i)} \leftarrow L_{k-1}^{(d,i)} \ell_k^{(d,i)}$    ▷ Update likelihood
            end for
            $\Phi_k^{(d)} \leftarrow quad(L_k^{(d,1)}, L_k^{(d,2)}, ..., L_k^{(d,q)})$    ▷ Compute likelihood approx., where $\Phi_k^{(d)} \approx p(\mathbf{y}_{1:k}|d)$, and $quad$ is a quadrature function, e.g. eq. (3.7) or (3.8)
        end for
        for destination $d \in \mathscr{D}$ do
            $u_d \leftarrow \frac{p(d)\Phi_k^{(d)}}{\sum_{j \in \mathscr{D}} p(d)\Phi_k^{(j)}}$    ▷ The probability of any given $d \in \mathscr{D}$
        end for
        return $u_d \approx p(d|\mathbf{y}_{1:k})$    ▷ Destination posterior after the $k^{\text{th}}$ measurement
    end for
end procedure
```

eq. (3.24b).

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) = \int_{t_f\in\mathscr{T}} \left(\sum_{d\in\mathscr{D}} p(\mathbf{x}_k|\mathbf{y}_{1:k},t_f,d)p(t_f|d)p(d)\right) dt_f \tag{3.24a}$$

$$\approx \sum_{i=1}^{q}\sum_{d\in\mathscr{D}} u_{d,i}\mathscr{N}\left(\mathbf{x}_k;\ [\mathbf{I},\mathbf{0}]\hat{\mathbf{z}}_k^{(d,i)},\ [\mathbf{I},\mathbf{0}]\mathbf{P}_k^{(d,i)}[\mathbf{I},\mathbf{0}]^T\right) \tag{3.24b}$$

with the weights

$$u_{d,i} = \frac{p(\mathbf{x}_k|t_{fi},d)p(t_{fi}|d)p(d)}{\sum_{i=1}^{q}\sum_{d\in\mathscr{D}} p(\mathbf{x}_k|t_{fi},d)p(t_{fi}|d)p(d)} \tag{3.25}$$

To predict future states of the tracked object, for a given destination d and arrival time t_{fi}, the predicting step in KF (step 2 and 3 in algorithm 1) can be used to calculate the predicted state and covariance for $t_\kappa > t_k$; where t_k is the current time and t_κ is a future time. These are then given by

$$\hat{\mathbf{z}}_{\kappa+1|k}^{(d,i)} = \mathbf{S}_\kappa^{(d,i)}\hat{\mathbf{z}}_{\kappa|k}^{(d,i)} + \tilde{\mathbf{b}}_\kappa^{(d,i)} \tag{3.26a}$$

$$\hat{\mathbf{P}}_{\kappa+1|k}^{(d,i)} = \mathbf{S}_\kappa^{(d,i)}\hat{\mathbf{P}}_{\kappa|k}^{(d,i)}\left(\mathbf{S}_\kappa^{(d,i)}\right)^T + \mathbf{U}_\kappa^{(d,i)} \tag{3.26b}$$

3.4 Bridging model using Bayesian filtering

Liang et al. [6] introduced a different approach to the bridging method described in section 3.3. Instead of using a joint state of the current and the final state, a pseudo-measurement $\tilde{\mathbf{y}}_f^{\{n\}}$ is introduced as the final measurement at destination d_n and arrival time t_f. The pseudo-measurement is normal distributed and is given as

$$p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_f,d_n) = \mathscr{N}(\tilde{\mathbf{y}}_f^{\{n\}};\tilde{\mathbf{H}}\mathbf{x}_f,\Sigma_f^{\{n\}}) \tag{3.27}$$

where $\mathbf{x}_f$ is the final state, and $\Sigma_f^{\{n\}}$ is the pseudo-measurement covariance at destination d_n. The pseudo-measurement matrix $\tilde{\mathbf{H}}$ might differ from the measurement matrix, and is determined by the available information at d_n. The pseudo-measurement introduces the conditioning on d_n, e.g. $p(\mathbf{y}_{1:k}|d_n) = p(\mathbf{y}_{1:k}|\tilde{\mathbf{y}}_f^{\{n\}})$. The measurements conditioned distribution can then be written with respect to $\tilde{\mathbf{y}}_f^{\{n\}}$ as

$$\begin{aligned} p(\mathbf{y}_{1:k}|d_n,t_f) &= p(\mathbf{y}_1|d_n,t_f)\prod_{m=2}^{k} p(\mathbf{y}_m|\mathbf{y}_{1:m-1},d_n,t_f) \\ &= p(\mathbf{y}_{1:k}|\tilde{\mathbf{y}}_f^{\{n\}},t_f) \end{aligned} \tag{3.28}$$

In the following, since the arrival time t_f is assumed known, it will not be included in the derivation where it does not play a role for the sake of brevity. Using Bayes' rule, the measurement-conditioned-state distribution can be written as

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1},\mathbf{y}_k,d_n) = p(\mathbf{y}_k|\mathbf{x}_k,\mathbf{y}_{1:k-1},d_n)p(\mathbf{x}_k|\mathbf{y}_{1:k-1},d_n) \tag{3.29}$$

Furthermore, the prediction error decomposition (PED) in eq. (3.28), $p(\mathbf{y}_k|\mathbf{y}_{1:k-1},d_n)$, can be found by integrating eq. (3.29) over $\mathbf{x}_k$, resulting in the following

$$p(\mathbf{y}_k|\mathbf{y}_{1:k-1},d_n) = \int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1},\tilde{\mathbf{y}}_f^{\{n\}})d\mathbf{x}_k \tag{3.30a}$$

$$= \int p(\mathbf{y}_k|\mathbf{x}_k)\frac{p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{y}_{1:k-1})}d\mathbf{x}_k \tag{3.30b}$$

where d_n is substituted with $\tilde{\mathbf{y}}_f^{\{n\}}$ and the second term under the integral in eq. (3.30a) is rewritten using the chain rule.[1]

As discussed in section 2.3.1, $p(\mathbf{x}_k|\mathbf{y}_{1:k-1})$ is the prediction step of the Bayes filter and for the linear system in eq. (3.1) it can be written on the following form.

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{y}_{1:k-1}) &= \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})d\mathbf{x}_{k-1} \\ &= \mathcal{N}(\mathbf{x}_k;\mu_{k|k-1},\Sigma_{k|k-1}) \end{aligned} \tag{3.31}$$

with the predicted mean and covariance as

$$\begin{aligned} \mu_{k|k-1} &= \mathbf{F}_T\mathbf{x}_{k-1} + \mathbf{m}_T \\ \Sigma_{k|k-1} &= \mathbf{F}_T\Sigma_{k-1}\mathbf{F}_T^T + \mathbf{Q}_T \end{aligned} \tag{3.32}$$

Similarly, the transition between the current state $\mathbf{x}_k$ and the pseudo-measurement $\tilde{\mathbf{y}}_f^{\{n\}}$ can be written as

$$\begin{aligned} p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_k) &= \int p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_f)p(\mathbf{x}_f|\mathbf{x}_k)d\mathbf{x}_f \\ &= \mathcal{N}(\tilde{\mathbf{y}}_f^{\{n\}};\mu_{\tilde{y}},\Sigma_{\tilde{y}}) \end{aligned} \tag{3.33}$$

with the mean and covariance as

$$\begin{aligned} \mu_{\tilde{y}} &= \tilde{\mathbf{H}}\left[\mathbf{F}_f\mathbf{x}_k + \mathbf{m}_f\right] \\ \Sigma_{\tilde{y}} &= \tilde{\mathbf{H}}\mathbf{Q}_f\tilde{\mathbf{H}}^T + \Sigma_f^{\{n\}} \end{aligned} \tag{3.34}$$

The two subscripts, T and f are used to differentiate the transition functions, where T represents that the transition matrix/vector is a function of the time step $T = t_k - t_{k-1}$ and f that it is a function of the time step $t_f - t_k$.

The product of the two Gaussian distributions in eqs. (3.31) and (3.33) result in the following distribution

$$p(\mathbf{x}_k|\mathbf{y}_{1:k-1},\tilde{\mathbf{y}}_f^{\{n\}}) \propto p(\mathbf{x}_k|\mathbf{y}_{1:k-1})p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_k) \tag{3.35a}$$

$$= \mathcal{N}(\mathbf{x}_k;\mu_{k|k-1},\Sigma_{k|k-1})\mathcal{N}(\tilde{\mathbf{y}}_f^{\{n\}};\mu_{\tilde{y}},\Sigma_{\tilde{y}}) \tag{3.35b}$$

$$= \mathcal{N}\left(\begin{bmatrix} \mathbf{x}_k \\ \tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f \end{bmatrix};\begin{bmatrix} \mu_{k|k-1} \\ \tilde{\mathbf{H}}\mathbf{F}_f\mu_{k|k-1} \end{bmatrix},\begin{bmatrix} \Sigma_{k|k-1} & \Sigma_{k|k-1}\mathbf{F}_f^T\tilde{\mathbf{H}}^T \\ \tilde{\mathbf{H}}\mathbf{F}_f\Sigma_{k|k-1} & \xi_* \end{bmatrix}\right) \tag{3.35c}$$

$$\propto \mathcal{N}(\mathbf{x}_k;\mu_*,\Sigma_*) \tag{3.35d}$$

[1]

$$p(A|B,C) = \frac{p(A,B,C)}{p(B,C)} = \frac{p(C|A,B)p(A,B)}{p(C,B)} = \frac{p(C|A,B)p(A|B)p(B)}{p(C|B)p(B)} = \frac{p(C|A,B)p(A|B)}{p(C|B)}$$

with the mean and covariance given by

$$\Sigma_* = \Sigma_{k|k-1} - \Sigma_{k|k-1}\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_*^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\Sigma_{k|k-1} \tag{3.36a}$$

$$\mu_* = \mu_{k|k-1} + \Sigma_{k|k-1}\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_*^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f - \tilde{\mathbf{H}}\mathbf{F}_f\mu_{k|k-1}\right) \tag{3.36b}$$

$$\xi_* := \Sigma_{\tilde{y}} + \tilde{\mathbf{H}}\mathbf{F}_f\Sigma_{k|k-1}\mathbf{F}_f^T\tilde{\mathbf{H}}^T \tag{3.36c}$$

Liang et al. [6] use the product identity in theorem 1 to compute the mean and covariance values (μ_* and Σ_*). Alternatively, one could use the product identity in theorem 2 which results in slightly different values shown in eq. (3.37). The covariance remains the same, whereas the mean becomes slightly different.

$$\Sigma_*^{alt.} = \Sigma_{k|k-1} - \Sigma_{k|k-1}\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_*^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\Sigma_{k|k-1} \tag{3.37a}$$

$$\mu_*^{alt.} = \Sigma_*\left(\Sigma_{k|k-1}^{-1}\mu_{k|k-1} + \mathbf{F}_f^T\tilde{\mathbf{H}}^T\Sigma_{\tilde{y}}^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f\right)\right) \tag{3.37b}$$

$$= \left(\mathbf{I} - \Sigma_{k|k-1}\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_*^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\right)\left(\mu_{k|k-1} + \Sigma_{k|k-1}\mathbf{F}_f^T\tilde{\mathbf{H}}^T\Sigma_{\tilde{y}}^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f\right)\right) \tag{3.37c}$$

Continuing the derivation, eq. (3.30) can now be written on the following form.

$$\begin{aligned} p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, d_n) &\propto \int \mathcal{N}(\mathbf{y}_k; \mathbf{H}\mathbf{x}_k, \mathbf{R}_k)\mathcal{N}(\mathbf{x}_k; \mu_*, \Sigma_*)d\mathbf{x}_k \\ &= \mathcal{N}(\mathbf{y}_k; \mu_y, \Sigma_y) \end{aligned} \tag{3.38}$$

$$\begin{aligned} \mu_y &= \mathbf{H}\mu_* \\ \Sigma_y &= \mathbf{H}\Sigma_*\mathbf{H}^T + \mathbf{R}_k \end{aligned} \tag{3.39}$$

Finally, this can then be used to find the likelihood of a measurement for a given destination and an arrival time using eq. (3.28).

For a given destination d and an arrival time t_{fi}, future states of a tracked object can be predicted using the predicting step of KF with mean and covariance from the *bridged* state estimate found in eq. (3.35d). For a future time $t_\kappa > t_k$ the predicted state and covariance are given by

$$\hat{\mathbf{x}}_{\kappa+1|k}^{(d,i)} = \mathbf{F}_\kappa^{(d,i)}\hat{\mathbf{x}}_{\kappa|k}^{(d,i)} + \mathbf{m}_\kappa^{(d,i)} \tag{3.40a}$$

$$\hat{\mathbf{P}}_{\kappa+1|k}^{(d,i)} = \mathbf{F}_\kappa^{(d,i)}\hat{\mathbf{P}}_{\kappa|k}^{(d,i)}\left(\mathbf{F}_\kappa^{(d,i)}\right)^T + (\Sigma_*)_\kappa^{(d,i)} \tag{3.40b}$$

where the transition functions with subscript κ are functions of the time step $t_{\kappa+1} - t_\kappa$. The a priori state distribution is given by

$$\hat{\mathbf{x}}_{k|k} \sim \mathcal{N}(\mu_*, \Sigma_*) \tag{3.41}$$

computed at the current time t_k.

3.5 Bridging model using Bayesian smoothing

Another approach of *bridging* the current state with the pseudo-measurement was also presented in [6], which uses Bayesian smoothing. Firstly, the equation for the backward update step of the

Bayesian smoother (eq. (2.26b) in section 2.3.2) is rewritten such that the transition goes from $\mathbf{x}_{k-1}$ to $\mathbf{x}_f$ instead of going to $\mathbf{x}_k$, i.e. in eq. (2.26b), $\mathbf{x}_k$ is replaced with $\mathbf{x}_{k-1}$, and $\mathbf{x}_{k+1}$ with $\mathbf{x}_f$, resulting in the following

$$\begin{aligned} p(\mathbf{x}_{k-1}|\mathbf{y}_{1:f}) =& \; p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \int \frac{p(\mathbf{x}_f|\mathbf{x}_{k-1})p(\mathbf{x}_f|\mathbf{y}_{1:f})}{p(\mathbf{x}_f|\mathbf{y}_{1:k-1})} d\mathbf{x}_f && (3.42a) \\ =& \; p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \int \frac{p(\mathbf{x}_f|\mathbf{x}_{k-1})}{p(\mathbf{x}_f|\mathbf{y}_{1:k-1})} p(\mathbf{x}_f|\mathbf{y}_{1:k-1},\tilde{\mathbf{y}}_f^{\{n\}}) d\mathbf{x}_f && (3.42b) \\ =& \; p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \int \frac{p(\mathbf{x}_f|\mathbf{x}_{k-1})}{p(\mathbf{x}_f|\mathbf{y}_{1:k-1})} p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_f,\mathbf{y}_{1:k-1}) p(\mathbf{x}_f|\mathbf{y}_{1:k-1}) d\mathbf{x}_f && (3.42c) \\ =& \; p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \int p(\mathbf{x}_f|\mathbf{x}_{k-1}) p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_f) d\mathbf{x}_f && (3.42d) \end{aligned}$$

where eq. (3.42a) is the backward update step with substituted variables. The first term in eq. (3.42d), $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1})$, is the output of a KF, and the integral is the transition between the previous state $\mathbf{x}_{k-1}$ and the pseudo-measurement $\tilde{\mathbf{y}}_f^{\{n\}}$. The transitional distribution is found in similar manner to eqs. (3.33) and (3.34), just with respect to time step t_{k-1} instead of t_k, and is given by

$$\begin{aligned} p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_{k-1}) &= \int p(\mathbf{x}_f|\mathbf{x}_{k-1}) p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_f) d\mathbf{x}_f \\ &= \mathcal{N}(\tilde{\mathbf{y}}_f^{\{n\}}; \mu_{\tilde{y}-1}, \Sigma_{\tilde{y}-1}) \end{aligned} \qquad (3.43)$$

with the mean and covariance as

$$\begin{aligned} \mu_{\tilde{y}-1} &= \tilde{\mathbf{H}}[\mathbf{F}_{f-1}\mathbf{x}_{k-1} + \mathbf{m}_{f-1}] \\ \Sigma_{\tilde{y}-1} &= \tilde{\mathbf{H}}\mathbf{Q}_{f-1}\tilde{\mathbf{H}}^T + \Sigma_f^{\{n\}} \end{aligned} \qquad (3.44)$$

where the $f-1$ subscript represents the transition function with respect to the time step $t_f - t_{k-1}$. Furthermore, writing the distribution $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:f})$ in eq. (3.42d) as $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1},\tilde{\mathbf{y}}_f^{\{n\}})$, the distribution can be written as

$$\begin{aligned} p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1},\tilde{\mathbf{y}}_f^{\{n\}}) =& \; p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_{k-1}) \\ =& \; \mathcal{N}(\mathbf{x}_{k-1}; \mu_{k-1}, \Sigma_{k-1}) \mathcal{N}(\tilde{\mathbf{y}}_f^{\{n\}}; \mu_{\tilde{y}-1}, \Sigma_{\tilde{y}-1}) \\ \propto& \; \mathcal{N}(\mathbf{x}_{k-1}; \tilde{\mu}, \tilde{\Sigma}) \end{aligned} \qquad (3.45)$$

with the mean and covariance given as

$$\begin{aligned} \tilde{\Sigma} &= \Sigma_{k-1} - \Sigma_{k-1}\mathbf{F}_{f-1}^T\tilde{\mathbf{H}}^T\tilde{\xi}^{-1}\tilde{\mathbf{H}}\mathbf{F}_{f-1}\Sigma_{k-1} && (3.46a) \\ \tilde{\mu} &= \mu_{k-1} + \Sigma_{k-1}\mathbf{F}_{f-1}^T\tilde{\mathbf{H}}^T\tilde{\xi}^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_{f-1} - \tilde{\mathbf{H}}\mathbf{F}_{f-1}\mu_{k-1}\right) && (3.46b) \\ \tilde{\mu}^{alt.} &= \tilde{\Sigma}\left(\Sigma_{k-1}^{-1}\mu_{k-1} + \mathbf{F}_{f-1}^T\tilde{\mathbf{H}}^T\Sigma_{\tilde{y}}^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_{f-1}\right)\right) && (3.46c) \\ \tilde{\xi} &= \Sigma_{\tilde{y}-1} + \tilde{\mathbf{H}}\mathbf{F}_{f-1}\Sigma_{k-1}\mathbf{F}_{f-1}^T\tilde{\mathbf{H}}^T && (3.46d) \end{aligned}$$

where the alternative mean, $\tilde{\mu}^{alt.}$ is computed using theorem 2 and $\tilde{\mu}$ is computed using theorem 1 as mentioned above. In the following the two mean values will be presented alongside each other where the alternative mean will have the superscript *alt.*

Secondly, the PED in eq. (3.30a) can be reformulated to include $p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1},\tilde{\mathbf{y}}_f^{\{n\}})$ from eq. (3.45). This leads to the following PED

$$
\begin{aligned}
p(\mathbf{y}_k|\mathbf{y}_{1:k-1},d_n) &= \int p(\mathbf{y}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{y}_{1:k-1},\tilde{\mathbf{y}}_f^{\{n\}})d\mathbf{x}_k && \text{(3.47a)}\\
&= \int p(\mathbf{y}_k|\mathbf{x}_k)\int p(\mathbf{x}_k|\mathbf{x}_{k-1},\tilde{\mathbf{y}}_f^{\{n\}})p(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1},\tilde{\mathbf{y}}_f^{\{n\}})d\mathbf{x}_{k-1}\;d\mathbf{x}_k && \text{(3.47b)}\\
&= \int \mathcal{N}(\mathbf{y}_k;\mathbf{H}\mathbf{x}_k,\mathbf{R}_k)\int \mathcal{N}(\mathbf{x}_k;\mu_\dagger,\Sigma_\dagger)\mathcal{N}(\mathbf{x}_{k-1};\tilde{\mu},\tilde{\Sigma})d\mathbf{x}_{k-1}\;d\mathbf{x}_k && \text{(3.47c)}
\end{aligned}
$$

Where $p(\mathbf{x}_k|\mathbf{x}_{k-1},\tilde{\mathbf{y}}_f^{\{n\}})$ in eq. (3.47b) can be written as

$$
\begin{aligned}
p(\mathbf{x}_k|\mathbf{x}_{k-1},\tilde{\mathbf{y}}_f^{\{n\}}) &\propto\; p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\tilde{\mathbf{y}}_f^{\{n\}}|\mathbf{x}_k)\\
&=\; \mathcal{N}(\mathbf{x}_k;\mu_{k|k-1},\mathbf{Q}_T)\mathcal{N}(\tilde{\mathbf{y}}_f^{\{n\}};\mu_{\tilde{y}},\Sigma_{\tilde{y}})\\
&\propto\; \mathcal{N}(\mathbf{x}_k;\mu_\dagger,\Sigma_\dagger)
\end{aligned}
\tag{3.48}
$$

with the mean and covariance given as

$$
\begin{aligned}
\Sigma_\dagger &= \mathbf{Q}_T - \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_\dagger^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\mathbf{Q}_T && \text{(3.49a)}\\
\mu_\dagger &= \mu_{k|k-1} + \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_\dagger^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f - \tilde{\mathbf{H}}\mathbf{F}_f\mu_{k|k-1}\right) && \text{(3.49b)}\\
\mu_\dagger^{alt.} &= \left(\mathbf{I} - \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_\dagger^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\right)\left(\mu_{k|k-1} + \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\Sigma_{\tilde{y}}^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \mathbf{H}\mathbf{m}_f\right)\right) && \text{(3.49c)}\\
\xi_\dagger &= \Sigma_{\tilde{y}} + \tilde{\mathbf{H}}\mathbf{F}_f\mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T && \text{(3.49d)}
\end{aligned}
$$

Furthermore, by substituting $\mu_{k|k-1}$ with $\mathbf{F}_T\mathbf{x}_{k-1} + \mathbf{m}_T$ in $\mu_\dagger$ in eq. (3.49), the inner integral in eq. (3.47c) can be solved as shown in eq. (3.50).

$$
\int \mathcal{N}(\mathbf{x}_k;\mu_\dagger,\Sigma_\dagger)\mathcal{N}(\mathbf{x}_{k-1};\tilde{\mu},\tilde{\Sigma})d\mathbf{x}_{k-1} = \;\mathcal{N}(\mathbf{x}_k;\mu_\ddagger,\Sigma_\ddagger)
\tag{3.50}
$$

with mean and covariance given as

$$
\begin{aligned}
\mu_\ddagger &= \mathbf{F}_T\tilde{\mu} + \mathbf{m}_T + \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_\dagger^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f - \tilde{\mathbf{H}}\mathbf{F}_f(\mathbf{F}_T\tilde{\mu} +^* \mathbf{m}_T)\right) && \text{(3.51a)}\\
\mu_\ddagger^{alt.} &= \left(\mathbf{I} - \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_\dagger^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\right)\left(\mathbf{F}_T\tilde{\mu} + \mathbf{m}_T + \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\Sigma_{\tilde{y}}^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f\right)\right) && \text{(3.51b)}\\
\Sigma_\ddagger &= \mathbf{A}_\ddagger\tilde{\Sigma}\mathbf{A}_\ddagger^T + \Sigma_\dagger && \text{(3.51c)}\\
\mathbf{A}_\ddagger &= \left(\mathbf{I} - \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_\dagger^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\right)\mathbf{F}_T && \text{(3.51d)}
\end{aligned}
$$

Finally, the PED (eq. (3.47)) using the smoothing approach is given as eq. (3.52).

$$
\begin{aligned}
p(\mathbf{y}_k|\mathbf{y}_{1:k-1},d_n) &= \int \mathcal{N}(\mathbf{y}_k;\mathbf{H}\mathbf{x}_k,\mathbf{R}_k)\mathcal{N}(\mathbf{x}_k;\mu_\ddagger,\Sigma_\ddagger)d\mathbf{x}_k\\
&= \mathcal{N}(\mathbf{y}_k;\mu_y,\Sigma_y)
\end{aligned}
\tag{3.52}
$$

with the mean and covariance given as

$$
\begin{aligned}
\Sigma_y &= \mathbf{H}\Sigma_\ddagger\mathbf{H}^T + \mathbf{R}_k\\
\mu_y &= \mathbf{H}\mu_\ddagger
\end{aligned}
\tag{3.53}
$$

For an overview of the Σ_y and μ_y values, these have been written out in eq. (3.54) to exclude Σ and μ values with subscript † and ‡.

$$\Sigma_y = \mathbf{H}\left[\mathbf{A}_{\ddagger}\tilde{\Sigma}\mathbf{A}_{\ddagger}^T + \mathbf{Q}_T - \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_{\dagger}^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\mathbf{Q}_T\right]\mathbf{H}^T + \mathbf{R}_k \tag{3.54a}$$

$$\mu_y = \mathbf{H}\left[\mathbf{F}_T\tilde{\mu} + \mathbf{m}_T + \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_{\dagger}^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f - \tilde{\mathbf{H}}\mathbf{F}_f(\mathbf{F}_T\tilde{\mu} +^* \mathbf{m}_T)\right)\right] \tag{3.54b}$$

$$\mu_y^{alt.} = \mathbf{H}\left(\mathbf{I} - \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_{\dagger}^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\right)\left[\mathbf{F}_T\tilde{\mu} + \mathbf{m}_T + \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\Sigma_{\tilde{y}}^{-1}\left(\tilde{\mathbf{y}}_f^{\{n\}} - \tilde{\mathbf{H}}\mathbf{m}_f\right)\right] \tag{3.54c}$$

$$\mathbf{A}_{\ddagger} = \left(\mathbf{I} - \mathbf{Q}_T\mathbf{F}_f^T\tilde{\mathbf{H}}^T\xi_{\dagger}^{-1}\tilde{\mathbf{H}}\mathbf{F}_f\right)\mathbf{F}_T \tag{3.54d}$$

$$\xi_{\dagger} = \Sigma_{\tilde{y}} + \tilde{\mathbf{H}}\mathbf{F}_f\mathbf{Q}(T)\mathbf{F}_f^T\tilde{\mathbf{H}}^T \tag{3.54e}$$

$$\Sigma_{\tilde{y}} = \tilde{\mathbf{H}}\mathbf{Q}_f\tilde{\mathbf{H}}^T + \Sigma_f^{\{n\}} \tag{3.54f}$$

An algorithm for destination inference for both the filtering approach, described in section 3.4, and the smoothing approach from this section, is given in algorithm 4.

[*] In [6], this sign was given out to be a minus, which seems to be a misprint where it should have been a plus sign. However, after testing, there was no visible difference by using plus or minus here. This is likely because $\mathbf{m}_T$ is substantially smaller than $\mathbf{F}_T\tilde{\mu}$ and these are then multiplied with high valued matrices, leading to a negligible vector.

Algorithm 4 Destination inference with Pseudo-Measurements, filtering and smoothing approach [6]

```
procedure DIPSEUDO($\mathbf{y}_{1:k}$, $\{\tilde{\mathbf{y}}_f^{\{n\}}, \forall d_n = d_1, ..., d_N\}$ )
    Initialize Set the mean and covariance a priori. Set initial likelihood $L_0^{(d,i)} = 1$
    for measurements $k = 1, ..., K$ do
        for destination $d \in \mathscr{D}$ do
            for quadrature point $i \in 1, ..., q$ do
                $\{\mu_{k|k-1}, \Sigma_{k|k-1}, \mu_k, \Sigma_k\}$
                    $\leftarrow KF(\mu_{k-1}, \Sigma_{k-1}, \mathbf{y}_k)$        ▷ Run KF iteration (algorithm 1) to compute predicted and corrected state and covariance
                Compute $\mu_*$ and $\Sigma_*$ in eq. (3.36)        ▷ Filtering approach
                Compute $\tilde{\mu}$ and $\tilde{\Sigma}$ in eq. (3.46)        ▷ Smoothing approach
                Compute $\ell_k^{(d,i)} \leftarrow p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, d, t_{fi})$ in eq. (3.38)        ▷ Filtering approach
                Compute $\ell_k^{(d,i)} \leftarrow p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, d, t_{fi})$ in eq. (3.52)        ▷ Smoothing approach
                $L_k^{(d,i)} \leftarrow L_{k-1}^{(d,i)} \ell_k^{(d,i)}$        ▷ Update likelihood
            end for
            $\Phi_k^{(d)} \leftarrow quad(L_k^{(d,1)}, L_k^{(d,2)}, ..., L_k^{(d,q)})$        ▷ Compute likelihood approx., where $\Phi_k^{(d)} \approx p(\mathbf{y}_{1:k}|d)$, and $quad$ is a quadrature function, e.g. eq. (3.7) or (3.8)
        end for
        for destination $d \in \mathscr{D}$ do
            $u_d \leftarrow \frac{p(d)\Phi_k^{(d)}}{\sum_{j \in \mathscr{D}} p(d)\Phi_k^{(j)}}$        ▷ The probability of any given $d \in \mathscr{D}$
        end for
        return $u_d \approx p(d|\mathbf{y}_{1:k})$        ▷ Destination posterior after the $k^{\text{th}}$ measurement
    end for
end procedure
```

Chapter 4

Motion models

This chapter presents the thee motion models that were used for the simulations. These are the constant velocity (CV) model, equilibrium reverting velocity (ERV) model, and Ornstein-Uhlenbeck (OU) model.

4.1 Constant velocity model

A commonly used model for estimating the motion of an object is the (nearly) constant velocity (CV) model. The object moves at a nearly constant velocity where the acceleration is modelled as white Gaussian noise with zero mean [12, pp. 269-270]. The stochastic differential equation (SDE) of the model, for a two dimensional case, is given in eq. (4.1) and the state vector is given by $\mathbf{x} = [\mathbf{p}^T,\ \mathbf{v}^T]^T$, where $\mathbf{p}$ is the position vector and $\mathbf{v}$ the velocity vector. Considering the two dimensional north-east coordinate system, the state vector in this case becomes $\mathbf{x} = [x\ y\ v_x\ v_y]^T$.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{G}\mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{q}\delta(t-\tau)) \tag{4.1a}$$

with matrices given as

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{q} = \begin{bmatrix} \sigma_a^2 & 0 \\ 0 & \sigma_a^2 \end{bmatrix} \tag{4.1b}$$

The elements in $\mathbf{q}$, σ_a, are the process noise variance that can be describe as the expected velocity change, i.e. the systems acceleration. Since the acceleration should be small compared to the actual velocity, a relatively small value of σ_a should be chosen. Discretization of the system with time step T is shown in eq. (4.2) [12, pp. 269-270].

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}\delta(t-\tau)) \tag{4.2a}$$

with matrices given as

$$\mathbf{F} = e^{\mathbf{A}T} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix} \sigma_a^2 \tag{4.2b}$$

4.2 Equilibrium reverting velocity

For an object in a motion where the final state is known, its motion can be modelled such that the object will revert towards its final state. Ahmad et al. [13] introduced a variation of the CV model, based on Ornstein-Uhlenbeck (OU) model, called equilibrium reverting velocity (ERV), for tracking and destination inference. The motion of a tracked object with a known destination can be described as it is being pulled by the destination $\mathbf{x}_d$ with a strength proportional to the distance between them, i.e. similar to a spring with a natural length of zero, connecting the two, and pulling them together. The SDE is given by

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x}_d - \mathbf{x}) + \mathbf{G}\mathbf{n}, \qquad \mathbf{n} \sim \mathscr{N}(\mathbf{0}, \mathbf{q}\delta(t-\tau)) \tag{4.3a}$$

where

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ \eta_x & 0 & \rho_x & 0 \\ 0 & \eta_y & 0 & \rho_y \end{bmatrix} \tag{4.3b}$$

and $\mathbf{x}_d = [x_d,\ y_d,\ 0,\ 0]$ is the position of the final destination with zero velocity. Drag coefficients $\eta \in \{0,1\}$ and mean reversion strengths $\rho \in \{0,1\}$ are introduced to the system matrix $\mathbf{A}$ to each spacial dimension for managing the “pulling” force. The process noise, $\mathbf{q}$, remains the same as for the CV model.

Integrating over the time interval T results in the following discretization

$$\mathbf{x}_{k+1} = \mathbf{F}_T\mathbf{x}_k + \mathbf{m}_T + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathscr{N}(\mathbf{0}, \mathbf{Q}_T\delta(t-\tau)) \tag{4.4a}$$

with parameters given as

$$\mathbf{F}_T = e^{-\mathbf{A}T}, \tag{4.4b}$$

$$\mathbf{m}_T = (\mathbf{I} - e^{-\mathbf{A}T})\mathbf{x}_d, \tag{4.4c}$$

$$\mathbf{Q}_T = \int_0^T e^{-\mathbf{A}(T-\tau)}\mathbf{G}\mathbf{q}\mathbf{G}^T e^{-\mathbf{A}^T(T-\tau)} d\tau \tag{4.4d}$$

The covariance matrix $\mathbf{Q}_T$ can be found by using Van Loan’s method [14]. Using this for the system shown in eq. (4.4d), the (partial) relationship between its matrices can be described as

$$\exp\left(\begin{bmatrix} \mathbf{A} & \mathbf{G}\mathbf{q}\mathbf{G}^T \\ \mathbf{0} & -\mathbf{A}^T \end{bmatrix} T\right) = \begin{bmatrix} \dots & \mathbf{F}_T^{-1}\mathbf{Q}_T \\ \mathbf{0} & \mathbf{F}_T^T \end{bmatrix} := \begin{bmatrix} \dots & \mathbf{V}_2 \\ \mathbf{0} & \mathbf{V}_1 \end{bmatrix} \tag{4.5}$$

From which, the covariance matrix $\mathbf{Q}$ can be found by first extracting $\mathbf{V}_1$ and $\mathbf{V}_2$, and then multiply these such that $\mathbf{Q} = \mathbf{V}_1^T\mathbf{V}_2$.

For a scenario where the drag and reversion strength coefficients are set to zero, the $\mathbf{F}$ matrix in eq. (4.4) becomes the same as for the CV model and the vector $\mathbf{m}$ becomes

$$\mathbf{m}_T = (\mathbf{I} - \mathbf{F}_T)\mathbf{x}_d = \begin{bmatrix} 0 & 0 & -T & 0 \\ 0 & 0 & 0 & -T \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_d \tag{4.6}$$

Additionally, if the final state at destination d, $\mathbf{x}_d$, is said to have zero velocity, the model becomes the same as the CV model. In this thesis, only the movement of watercraft are considered and since its movement do not behave similar to a spring, the drag and reversion coefficients are assumed zero.

4.3 Ornstein-Uhlenbeck model

Similarly to the ERV model, an OU process can be used to describe the motion of an object. Instead of assuming that an object in motion reverts towards a destination, the velocity will revert towards a mean velocity. The OU model was presented in [15], and its SDE is given as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}v + \mathbf{G}\mathbf{n}, \qquad \mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{q}\delta(t-\tau)) \tag{4.7a}$$

where v is the mean velocity vector and the matrices are given as

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{0} & -\Theta \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \Theta \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \tag{4.7b}$$

The Θ matrix quantifies the mean reversion effect, i.e. at which rate the velocity will revert to the desired value. Θ is assumed to have positive and distinct eigenvalues such that it can be written as

$$\Theta \; = \; \mathbf{E}\Gamma\mathbf{E}^{-1} \tag{4.8}$$

where $\mathbf{E}$ contains the eigenvectors, and Γ contains the eigenvalues on the diagonal. For simplicity $\mathbf{E}$ is assumed to be the identity matrix such that $\Theta \; = \; \Gamma \; = \; \text{diag}(\gamma)$. Integrating eq. (4.7) over the time interval T results in the following discretization

$$\mathbf{x}_{k+1} = \; \mathbf{F}(T,\gamma)\mathbf{x}_k + \mathbf{M}(T,\gamma)v_k + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_T\delta(t-\tau)) \tag{4.9a}$$

with matrices given as

$$\mathbf{F}(T,\gamma) = \begin{bmatrix} \mathbf{I} & (\mathbf{I} - e^{-\Gamma\, T})\Gamma^{-1} \\ \mathbf{0} & e^{-\Gamma\, T} \end{bmatrix}, \quad \mathbf{M}(T,\gamma) = \begin{bmatrix} t\mathbf{I} - (\mathbf{I} - e^{-\Gamma\, T})\Gamma^{-1} \\ \mathbf{I} - e^{-\Gamma\, T} \end{bmatrix} \tag{4.9b}$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{M}$ is the control input, both being a function of the time step and reversion values. The prediction covariance is given by

$$\mathbf{P}_{k+1} \;=\; \mathbf{F}(T,\gamma)\,\mathbf{P}_k\,\mathbf{F}(T,\gamma)^T + \Sigma_1 \circ \Sigma_2(T_k) \tag{4.10}$$

where the $\circ$ operator is the Hadamard product [1] and the Σ-covariance matrices are given by

$$\Sigma_1 = \begin{bmatrix} \frac{\sigma_x^2}{\gamma_x^3} & \frac{\sigma_{xy}}{\gamma_x\gamma_y} & \frac{\sigma_x^2}{2\gamma_x^2} & \frac{2\sigma_{xy}}{\gamma_x} \\ \frac{\sigma_{xy}}{\gamma_x\gamma_y} & \frac{\sigma_y^2}{\gamma_y^3} & \frac{2\sigma_{xy}}{\gamma_y} & \frac{\sigma_y^2}{2\gamma_y^2} \\ \frac{\sigma_x^2}{2\gamma_x^2} & \frac{2\sigma_{xy}}{\gamma_y} & \frac{\sigma_x^2}{\gamma_x} & \frac{2\sigma_{xy}}{\gamma_x+\gamma_y} \\ \frac{2\sigma_{xy}}{\gamma_x} & \frac{\sigma_y^2}{2\gamma_y^2} & \frac{2\sigma_{xy}}{\gamma_x+\gamma_y} & \frac{\sigma_y^2}{\gamma_y} \end{bmatrix} \tag{4.11a}$$

$$\Sigma_2(T_k) = \begin{bmatrix} f(T_k\gamma_x) & h(T_k,\gamma) & k(T_k\gamma_x) & g_1^*(T_k,\gamma) \\ h(T_k,\gamma) & f(T_k\gamma_y) & g_2^*(T_k,\gamma) & k(T_k\gamma_y) \\ k(T_k\gamma_x) & g_2^*(T_k,\gamma) & g(T_k\gamma_x) & g((\gamma_x+\gamma_y)T_k/2) \\ g_1^*(T_k,\gamma) & k(T_k\gamma_y) & g((\gamma_x+\gamma_y)T_k/2) & g(T_k\gamma_y) \end{bmatrix} \tag{4.11b}$$

with functions defined as

$$f(t) := \frac{1}{2}\left(2t + 4e^{-t} - e^{-2t} - 3\right) \tag{4.12a}$$

$$g(t) := \frac{1}{2}\left(1 - e^{-2t}\right) \tag{4.12b}$$

$$k(t) := e^{-2t}\left(1 - e^{-t}\right)^2 \tag{4.12c}$$

$$h(t,\gamma) := t - \frac{1-e^{-t\gamma_x}}{\gamma_x} - \frac{1-e^{-t\gamma_y}}{\gamma_y} + \frac{1-e^{-t(\gamma_x+\gamma_y)}}{\gamma_x+\gamma_y} \tag{4.12d}$$

$$g_1^*(t,\gamma) := \frac{g(\gamma_y T_k/2)}{\gamma_y} - \frac{g((\gamma_x+\gamma_y)T_k/2)}{\gamma_x+\gamma_y} \tag{4.12e}$$

$$g_2^*(t,\gamma) := \frac{g(\gamma_x T_k/2)}{\gamma_x} - \frac{g((\gamma_x+\gamma_y)T_k/2)}{\gamma_x+\gamma_y} \tag{4.12f}$$

To predict the motion of an object where the mean velocity is unknown, a dynamic mean velocity can be estimated using the n last measurements. In this thesis the average velocity of the last 10 measurements are used as the mean velocity, v_k, for the simulations.

For more details about the equations the reader is referred to [15].

[1] Example of Hadamard product:

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \circ \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1b_1 & a_2b_2 \\ a_3b_3 & a_4b_4 \end{bmatrix}$$

Chapter 5

Test scheme

This chapter describes the details about the test scheme. It provides information about the different scenarios tested and how AIS data was used as measurements. Thereafter, a description of how the system's parameters were tuned is presented.

5.1 Scenarios

The map area considered for all test runs was chosen to be the Trondheim fjord in Norway. The area in question is illustrated in fig. 5.1, along with the destinations considered. These are represented as covariance ellipses of possible final destinations for a tracked object. These ellipses were manually chosen such that their size and form felt natural for each destination. Some are destinations where the final velocity is expected to be zero, e.g. ports and harbours. Other destinations are places where the object is expected to pass through with velocity greater than zero, i.e. places where the object can leave the area, or exits.

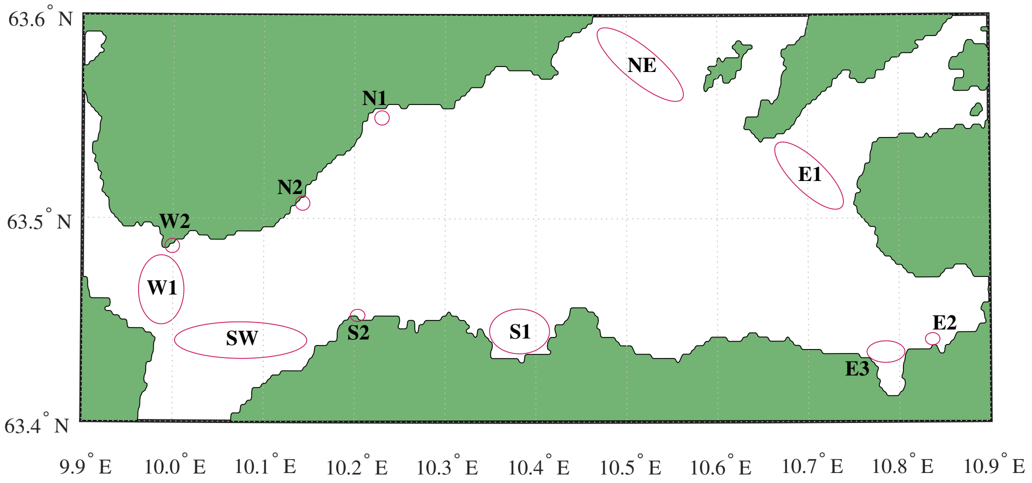


Figure 5.1: Map of the Trondheim fjord with the possible final destinations, represented as ellipses

The destination names and the centre of their location are listed in table 5.1. The destination *S1*, the area outside Trondheim, is a special case where it is considered as one area, but in reality it contains several possible destinations. Furthermore, there is relatively high activity within the area, and therefore it would be practical to consider the area on its own for destination inference and future trajectory prediction. To test the algorithm performance, five scenario examples were extracted from a collection of real-world AIS data. This collection is the same as was used in the study by Dalsnes et al. [16] provided by DNV GL. The scenarios are listed in table 5.2 along with the ship's name and type. The scenarios are labelled as their starting position and their final destination, e.g. *NE-S1* is the scenario where the tracked object is travelling from *NE* towards *S1*. The trajectory for each scenario is provided in chapter 6, where the results from the tests are presented.

Table 5.1: List of possible final destinations

Code	Name	Centre of ellipse $[lat, lon]$	Type
S1	Trondheim	$[63.4446, 10.3820]$	Port area
S2	Flakk	$[63.4524, 10.2038]$	Port
SW	South-West exit	$[63.4400, 10.0751]$	Exit
W1	West exit	$[63.4648, 9.9873]$	Exit
W2	Sandbakken	$[63.4864, 9.9991]$	Mariana
N2	Rorvik	$[63.5075, 10.1427]$	Port
N1	Vanvikan	$[63.5496, 10.2300]$	Marina
NE	North-East exit	$[63.5759, 10.5153]$	Exit
E1	East exit	$[63.5210, 10.7014]$	Exit
E2	Muruvik	$[63.4404, 10.8366]$	Marina
E3	Malvik	$[63.4342, 10.7851]$	Port area

Table 5.2: List of scenarios of ships travelling in the Trondheim fjord, along with the ships information. The ship types were retrieved from [17] using the code from the AIS

No.	Scenarios	Ship's name	Length $[m]$	Width $[m]$	Type $\{code\}$
1	*NE-S1*	Fosna Triton	15	5	Diving ops $\{34\}$
2	*W1-S1*	Kong Harald	122	20	Passenger $\{60\}$
3	*W1-E2*	Vestland	88	13	Cargo $\{70\}$
4	*N2-S2*	Lagatun	108	18	Passenger $\{69\}$
5	*NE-W1*	Ro Chief	53	12	Cargo $\{70\}$

5.2 Using AIS measurements

AIS is a worldwide system, extensively used to identify and track maritime vessels. A signal is transmitted from a vessel, using a small transponder fitted on the vessel, to coastal authorities and other ships automatically. The signal includes both static and dynamic information about the vessel, e.g. identification number, type of vessel, position, course over ground (COG), speed over ground (SOG), heading, etc. It is required, by international maritime law, for all international voyaging ships weighing 300 tonnes or more, and all passenger ships, to be fitted with AIS. [18]

AIS data was used as measurements for tracking a ship. Normally, position is from GNSS-measurements, and velocity comes from both COG and SOG measurements, which are usually derived from the position. Depending on sensors on board, these can be noisy and/or with large variance. Moreover, COG could oscillate when sailing over waves, but for some vessels these oscillations can be filtered out, which results on more stable COG measurements but at the cost of having larger variance. Additionally, COG is very unstable at low velocities, but SOG can be accurate also at low velocities. This is because when a vessel is almost stationary its positional measurements jump around the true position and the COG will follow. This does not affect the SOG as much since the measured position is random within a certain area which zeros out the speed. Typical heading measurements come from a different sensor, a gyroscope, that measures the change in orientation.

The measured position is given in latitude and longitude ($[\lambda, \varphi]$) and is converted to Cartesian coordinates using the MATLAB's built-in function `latlon2local` [19]. The velocity is computed from the SOG and COG measurements using trigonometry, as shown in eq. (5.2). The last measurement used in this thesis is the heading angle, denoted as ψ, is used directly. The measurement vector in geographic coordinates is shown in eq. (5.1a) and the corresponding Cartesian vector is shown in eq. (5.1b), where x and y are the positions in east and north direction, respectively.

$$\mathbf{y}^g = \left[\lambda,\ \varphi,\ \psi,\ \text{SOG},\ \text{COG}\right]^T \tag{5.1a}$$

$$\mathbf{y}^c = \left[x,\ y,\ \psi,\ \dot{x},\ \dot{y}\right]^T \tag{5.1b}$$

$$\dot{x} = \text{SOG}\sin(\text{COG}) \tag{5.2a}$$

$$\dot{y} = \text{SOG}\cos(\text{COG}) \tag{5.2b}$$

In the following, a measurement vector will always refer to a measurement vector in the Cartesian coordinate system.

5.3 State and measurement vectors

Two different state and measurement vectors were considered. The first containing the position and their derivatives, and the second containing the position, the course angle, and their derivatives. In the following, these will have the superscript a and b to represent the first and second, respectively.

The state vectors (**x**) and their corresponding measurement vectors (**y**) are shown in eq. (5.3).

$$\mathbf{x}^a = \begin{bmatrix} x, \ y, \ \dot{x}, \ \dot{y} \end{bmatrix}^T \tag{5.3a}$$

$$\mathbf{y}^a = \begin{bmatrix} x, \ y, \ \dot{x}, \ \dot{y} \end{bmatrix}^T \tag{5.3b}$$

$$\mathbf{x}^b = \begin{bmatrix} x, \ y, \ \psi, \ \dot{x}, \ \dot{y}, \ \dot{\psi} \end{bmatrix}^T \tag{5.3c}$$

$$\mathbf{y}^b = \begin{bmatrix} x, \ y, \ \psi, \ \dot{x}, \ \dot{y} \end{bmatrix}^T \tag{5.3d}$$

This leads to the following measurement and covariance matrices

$$\mathbf{H}^a = \mathbf{I}_4 \tag{5.4a}$$

$$\mathbf{H}^b = [\mathbf{I}_5, \ \mathbf{0}_{1\times 5}] \tag{5.4b}$$

$$\mathbf{R}^a = \begin{bmatrix} \sigma_p^2 & 0 & 0 & 0 \\ 0 & \sigma_p^2 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} + \frac{1}{720} \begin{bmatrix} v_x^2 & 0 & 0 & 0 \\ 0 & v_y^2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.5a}$$

$$\mathbf{R}^b = \begin{bmatrix} \sigma_p^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_p^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_h^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_v^2 \end{bmatrix} + \frac{1}{720} \begin{bmatrix} v_x^2 & 0 & 0 & 0 & 0 \\ 0 & v_y^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{5.5b}$$

where σ_p, σ_v and σ_h are the variances for position, velocity and heading angle, respectively. The covariance matrix **R** is a combination of two matrices, the first being the covariance matrix containing the variance on the diagonal. The variance is assumed to be the same in both x and y directions. The second matrix is the added error of the measured position which is proportional to the velocity (v_x and v_y), scaled with a suitable variance. In [20] the quantization error of the measurement is said to be uniform in ± 0.5 s, leading to the choice of a moment-matched zero mean Gaussian distribution with a standard deviation $\sigma = 1/\sqrt{12}$. Using minutes as the unit of time measurement results in $\sigma = 1/\sqrt{720}$, which is the scaling factor shown in eq. (5.5), i.e. σ^2.

In this thesis the unit used for time was chosen to be minutes instead of seconds. This was needed for computational stability. The possible arrival times were relatively large, up to 7 hours, and some transition matrices are a function of the time step between the current and the arrival time. This leads to very high valued matrices. Moreover, MATLAB was used for the computations and when it tried to work with these matrices it deemed it unable to work within a reasonable precision, which lead to singular matrices. This issue was solved by simply using minutes as the unit of time.

5.3.1 Tuning of covariance

For a filter to be considered consistent, its errors, on average, should be unbiased and have magnitude corresponding to the covariance calculated by the filter. This leads to the consistency analysis with the following criteria [12, pp. 234-236]

1. The state errors should have zero mean
2. The state errors should have magnitudes corresponding to the state covariance calculated by the filter
3. The innovation should have zero mean
4. The innovation should have magnitudes corresponding to the innovation covariance calculated by the filter
5. The innovation should be acceptable as white

Checking if criteria 2 and/or 4 holds is a typical test for consistency. To check if criteria 2 holds the normalised estimation error squared (NEES) is calculated at each iteration k.

$$\varepsilon_k^x = (\hat{\mathbf{x}}_k - \mathbf{x}_k)^T \mathbf{P}_k^{-1} (\hat{\mathbf{x}}_k - \mathbf{x}_k) \tag{5.6}$$

Where $\mathbf{x}$ is the true state, and $\hat{\mathbf{x}}$ and $\mathbf{P}$ are the filters state estimate and covariance, respectively. To check if criteria 4 holds the normalised innovation squared (NIS) is calculated at each iteration.

$$\varepsilon_k^y = (\hat{\mathbf{y}}_k - \mathbf{y}_k)^T \mathbf{J}_k^{-1} (\hat{\mathbf{y}}_k - \mathbf{y}_k) \tag{5.7}$$

This can be used to tune the process and measurement covariance, such that the error fits within a certain confidence interval, e.g. the average of NEES or NIS after N Monte Carlo simulations should fit within a 95% confidence interval. For a system where the true state $\mathbf{x}$ is unknown, one could check if criteria 4 holds [12, pp. 234-236]. In the following, only criteria 4 is considered since the true states are unknown.

The process variance (σ_a and σ_ψ) and the measurement variance (σ_p, σ_v and σ_h) were tuned for the different scenarios (ships). The tuning was perfomed by running the measurements through a KF using CV model for the two different state vectors, $\mathbf{x}^a$ and $\mathbf{x}^b$, and using OU model (with $\mathbf{x}^a$). The measurement covariance was assumed to be the same for all ships except *NE-S1*, since these were all large ships with similar velocity profile. Whereas, *NE-S1* was a smaller boat with higher accelerations and more noisier measurements. The parameters used for the tests are shown in table 5.3, along with the NIS limits and its computed average for each case. The limits were computed with the build-in MATLAB function `chi2inv` [21], using 95% confidence.

Table 5.3: The parameters used for the tests with the resulting average NIS. It is desired to have the average NIS between the NIS limits.

		Parameter						NIS		
		σ_p	σ_v	σ_h	σ_a	σ_ψ	γ	lower limit	upper limit	average
unit		m	$^{m}/_{min}$	deg	$^{m}/_{min^2}$	$^{deg}/_{min}$	min^{-1}			
NE-S1	$\mathbf{x}^a$	5	24	-	28	-	-	3.76	4.25	**3.92**
	$\mathbf{x}^b$	5	24	3	28	1.2	-	5.70	6.30	**6.07**
	OU	5	24	-	28	-	0.05	3.76	4.25	**3.94**
W1-S1	$\mathbf{x}^a$	2	3	-	9	-	-	3.79	4.21	**4.05**
	$\mathbf{x}^b$	2	3	1	9	0.7	-	5.75	6.26	**5.97**
	OU	2	3	-	9	-	0.05	3.79	4.21	**4.06**
W1-E2	$\mathbf{x}^a$	2	3	-	11	-	-	3.82	4.19	**3.98**
	$\mathbf{x}^b$	2	3	1	11	1.4	-	5.78	6.23	**5.97**
	OU	2	3	-	11	-	0.05	3.82	4.19	**3.97**
N2-S2	$\mathbf{x}^a$	2	3	-	14	-	-	3.54	4.49	**3.73**
	$\mathbf{x}^b$	2	3	1	14	1.5	-	5.43	6.60	**6.17**
	OU	2	3	-	14	-	0.05	3.54	4.49	**3.88**
NE-W1	$\mathbf{x}^a$	2	3	-	3	-	-	3.72	4.29	**3.75**
	$\mathbf{x}^b$	2	3	1	3	0.025	-	5.66	6.35	**5.76**
	OU	2	3	-	3	-	0.05	3.72	4.29	**3.81**

Chapter 6

Results

This chapter presents the results from testing the bridging methods described in chapter 3 using real measurements. Firstly, a short description is provided of the different combinations of bridging models and motion models (presented in chapter 4) considered in this thesis. Secondly, the results from the destination inference are presented for each scenario. Lastly, the results from the future prediction computations are presented.

6.1 Model cases

The bridging models described in sections 3.3 to 3.5 were used for the destination intent estimations. These are

1. (BF1) - Bayesian filtering using only the current state (section 3.4)
2. (BF2) - Bayesian filtering using the joint state $\mathbf{z}_k = [\mathbf{x}_k; \mathbf{x}_f]$ (section 3.3)
3. (BS) - Bayesian smoothing using only the current state (section 3.5)

In the following, these models will be referred as BF1, BF2 and BS as indicated in the parenthesis above. Additionally, to study the effect of including the heading angle as a state variable, both $\mathbf{x}^a$ and $\mathbf{x}^b$ were used in the ERV model. These were then compared, along with the OU model (with $\mathbf{x}^a$). The Bayesian bridging model along with the motion model, will be referred to as model cases, or simply just as cases. The combinations of these model cases are given in table 6.1.

Note that, for the setting in this thesis, the ERV model is strictly speaking a bridged CV model, since the drag and reversion strength coefficients are set to zero (see section 4.2).

6.2 Destination inference

For estimating the destination intent, a destination likelihood is computed at each time step, i.e. the destination weights, where the sum of the weights of all destinations are equal to one. How these weights develop over time are then compared for each model case.

Table 6.1: The different model cases tested and compared

Abbr.	Description
BF1-ERVa	BF1 bridging using ERV model with $\mathbf{x}^a$
BF2-ERVa	BF2 bridging using ERV model with $\mathbf{x}^a$
BS-ERVa	BS bridging using ERV model with $\mathbf{x}^a$
BF1-ERVb	BF1 bridging using ERV model with $\mathbf{x}^b$
BF2-ERVb	BF2 bridging using ERV model with $\mathbf{x}^b$
BS-ERVb	BS bridging using ERV model with $\mathbf{x}^b$
BF1-OU	BF1 bridging using OU model with $\mathbf{x}^a$
BF2-OU	BF2 bridging using OU model with $\mathbf{x}^a$
BS-OU	BS bridging using OU model with $\mathbf{x}^a$

Regarding the use of the mean μ given in [6], compared with the use of the alternative mean $\mu^{alt.}$, as discussed in sections 3.4 and 3.5. The use of these two produced the same results for all model cases (or at least with insignificant difference) except for BS-OU. Using OU model with the given Bayesian smoothing equations resulted in unexpected destination weights development. This is further discussed in section 6.2.6, where a comparison is presented. That being said, in sections 6.2.1 to 6.2.5 only the alternative mean will be considered.

In the following, for notational brevity, for cases where the use of either ERVa or ERVb produced the same results, the notation ERVa&b will be used to represent both ERVa and ERVb.

6.2.1 *NE-S1*: Destination inference for a boat travelling from *NE* to *S1*

The boat travelling from NE to S1, with trajectory shown in fig. 6.1, is a smaller vessel compared with the other ships discussed below. Hence, it's position, SOG and COG is more affected by waves and other external forces, leading to noisier measurements. Additionally, smaller boats tend to have higher directional and angular acceleration.

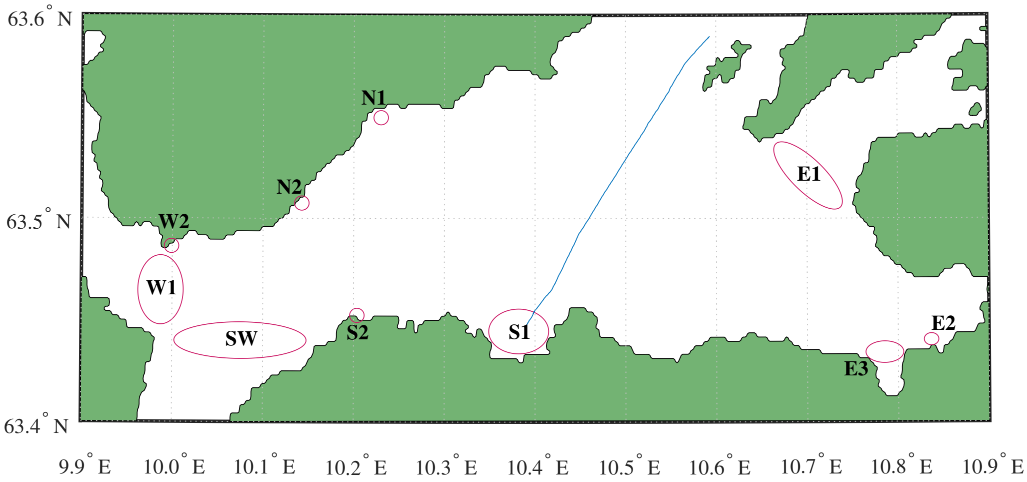
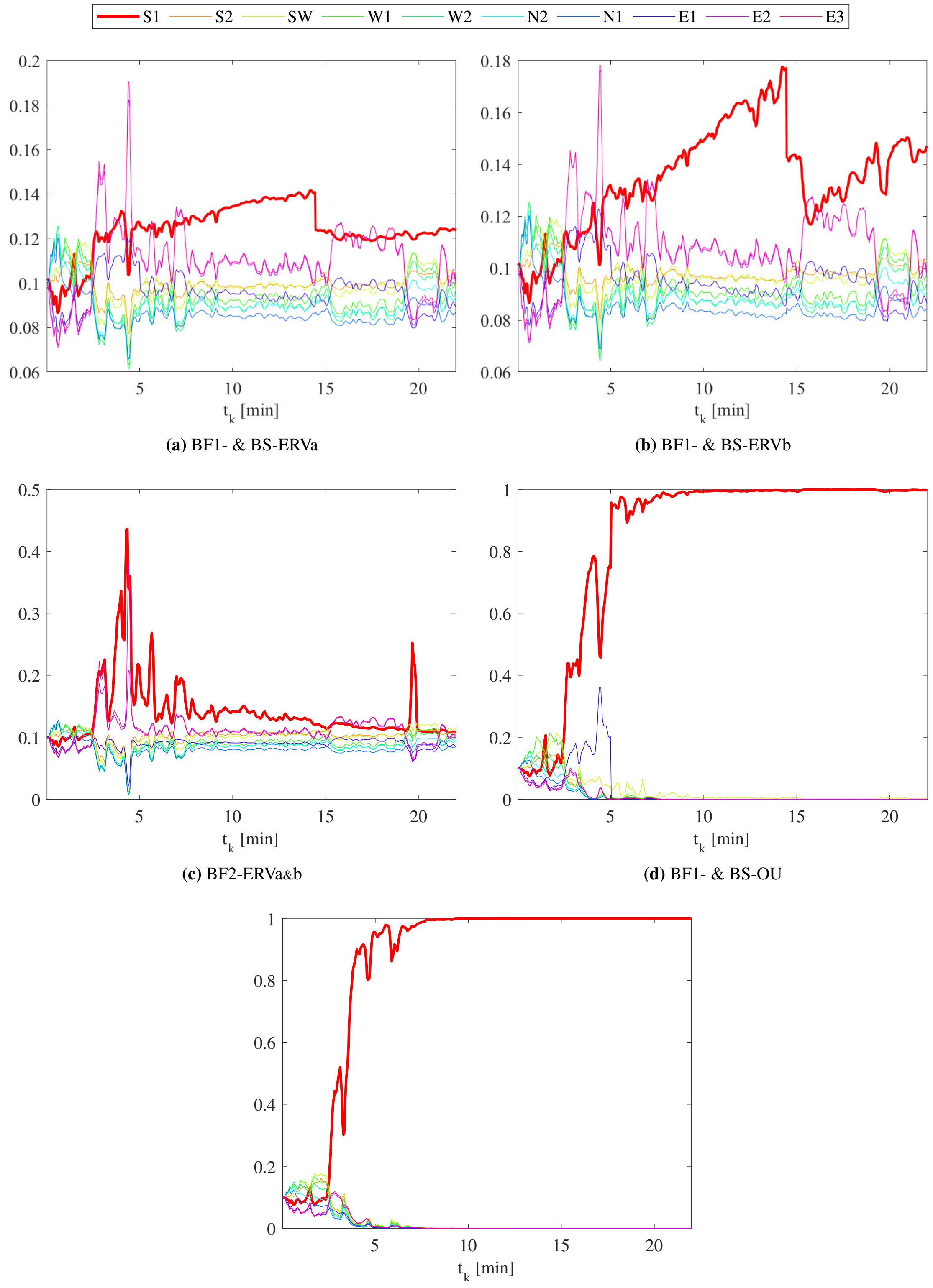


Figure 6.1: Measured trajectory of *Fosna Triton*, a diving ops. boat, travelling from NE to S1.

The computed weights for each destination over time is illustrated in fig. 6.2. The weights for the true destination, *S1*, is illustrated as a bold line for easier distinction. Many cases resulted in the



(a) BF1- & BS-ERVa

(b) BF1- & BS-ERVb

(c) BF2-ERVa&b

(d) BF1- & BS-OU

(e) BF2-OU

Figure 6.2: Computed destination weights (*y-axis*) at each time step, t_k, over the travelling time for the boat travelling from *NE* to *S1*. The destination weights for the true destination is illustrated as a bold line.

same, or very similar, results. There was insignificant difference in using either BF1 or BS, but using different motion models with these resulted in different destination weight development. For instance, using OU model resulted in much higher confidence for the true destination. Additionally, with BF2 the computed weights were the same when combined with ERVa and ERVb.

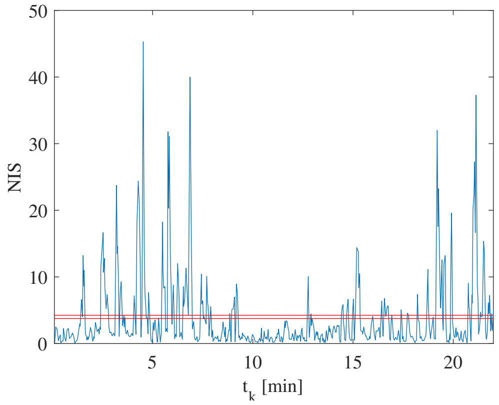
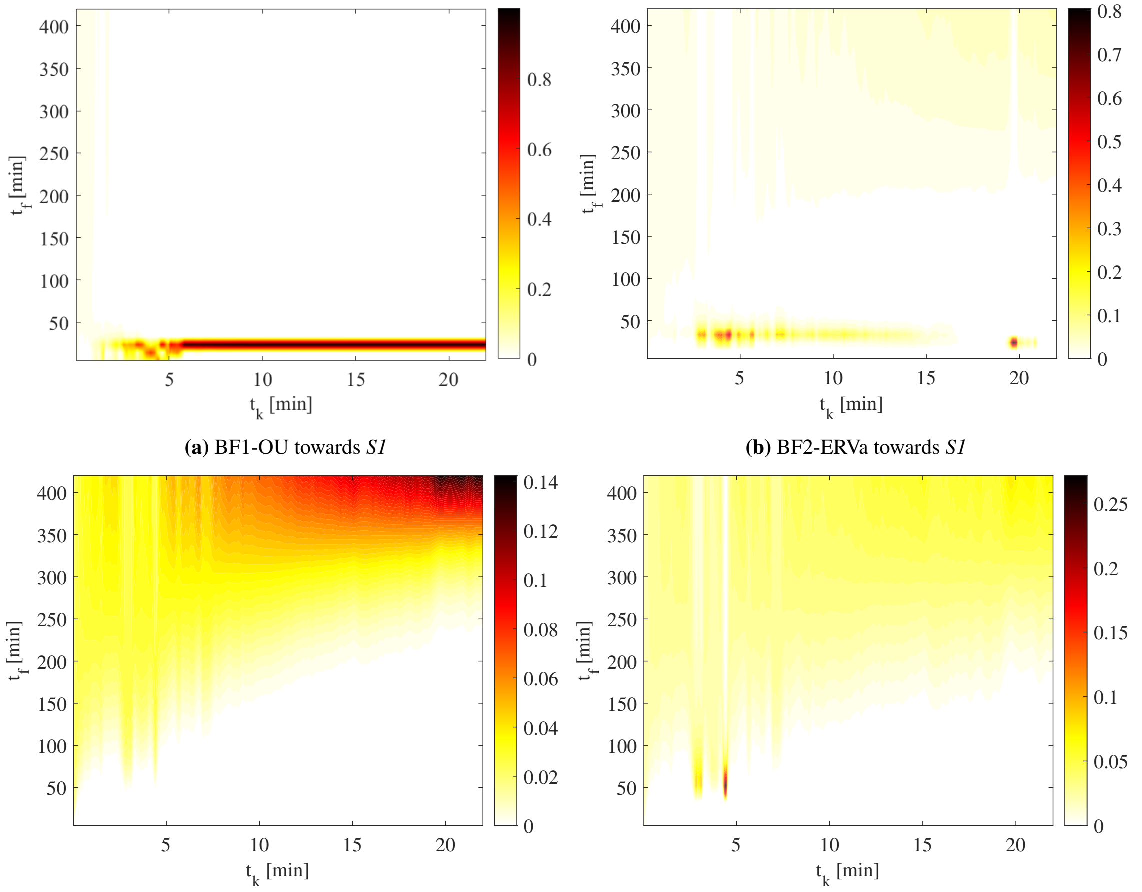


Figure 6.3: Computed NIS at each time step for *NE*-*S1* using BF1-ERVa. The red lines represent the desired values of the average NIS.

Regarding the measured trajectory, after about 2.5 minutes the boat changed its heading and starts heading towards *S1*. In fig. 6.1, this heading change might be hard to see because of the vast map area, but this is however a significant change. At that time a spike in the destination weights occurred for *S1* and the *E*-destinations. A similar phenomenon happened after 15 minutes, where the boat changes its heading slightly away from *S1* and the weights on *S1* dropped. This latter change in heading is most likely due to an island, Munkholmen, that is located north of S1 (not shown on



(a) BF1-OU towards *S1*

(b) BF2-ERVa towards *S1*

(c) BF1-OU towards *E2*

(d) BF2-ERVa towards *E2*

Figure 6.4: Arrival time (t_f) probability distribution development over the travelling time (t_k) for the boat travelling from *NE* to *S1*. Computed for the possible destinations *S1* and *E2*, using the model cases BF1-OU and BF2-ERVa. The values on the bar going from light to dark represent the probability values.

the map). At these two events the NIS values increased dramatically and consequently the model gets overconfident and trusts its estimates much more than the measurements. Initially, BF2-ERVa was most affected by this change but decreased its *S1*-weight over time, towards similar values as BF1-ERVa after the initial spike at t_k ~2.5 min. An example of the computed NIS for BF1-ERVa at each time step is illustrated in fig. 6.3.

Comparing figs. 6.2a and 6.2b the effect of using ERVa and ERVb with BF1 (and BS) can be seen. Both had a very similar trend where the destination MAP estimate was the same at most time instances, i.e. *S1*. However, by including the heading angle as a state variable (ERVb), the model got more confident the longer the heading was kept towards *S1*.

Using OU the destination weights for *S1* increased rapidly after the initial heading change and got extremely confident, arguably too confident, in its estimates. This shows how the model, using OU, will consider the object's heading towards a certain destination with high degree.

Furthermore, since the destination distribution is computed from the integration of the arrival time distribution, it is of interest to take a look on how the arrival time distribution develops over time. Figure 6.4 shows examples of how the arrival time distribution develops over time for the destinations *S1* and *E2* using BF1-OU and BF2-ERVa. It shows that both models had a good estimates of the arrival time at *S1* early on, but with different certainty. BF1-OU has a high certainty of its arrival time estimation, whereas BF2-ERVa has a decreasing certainty and becomes relatively uncertain between t_k=15 and 20 min.

6.2.2 *W1*-*S1*: Destination inference for a ship travelling from *W1* to *S1*

The ship travelling from *W1* to *S1*, with trajectory shown in fig. 6.5, is a large passenger ship called *Kong Harald*. It had a relatively constant velocity profile for the majority of its travel.

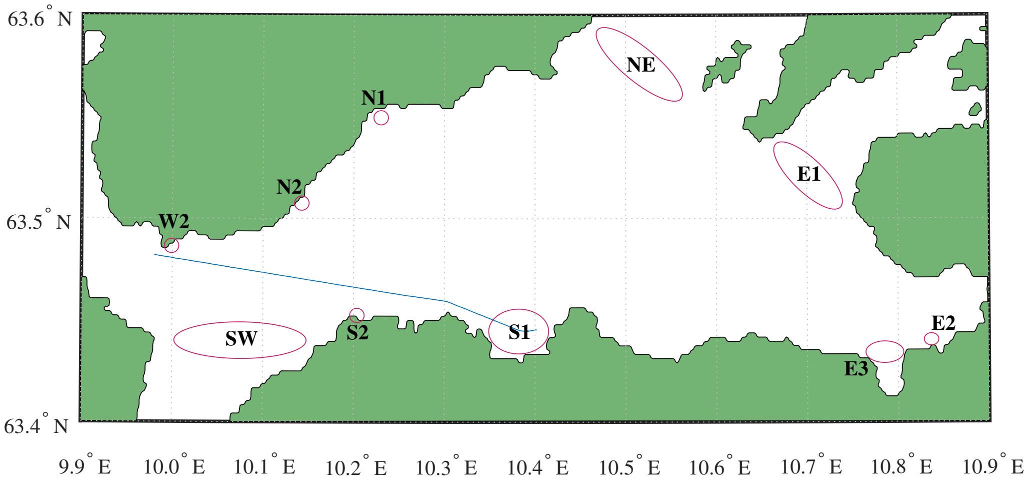
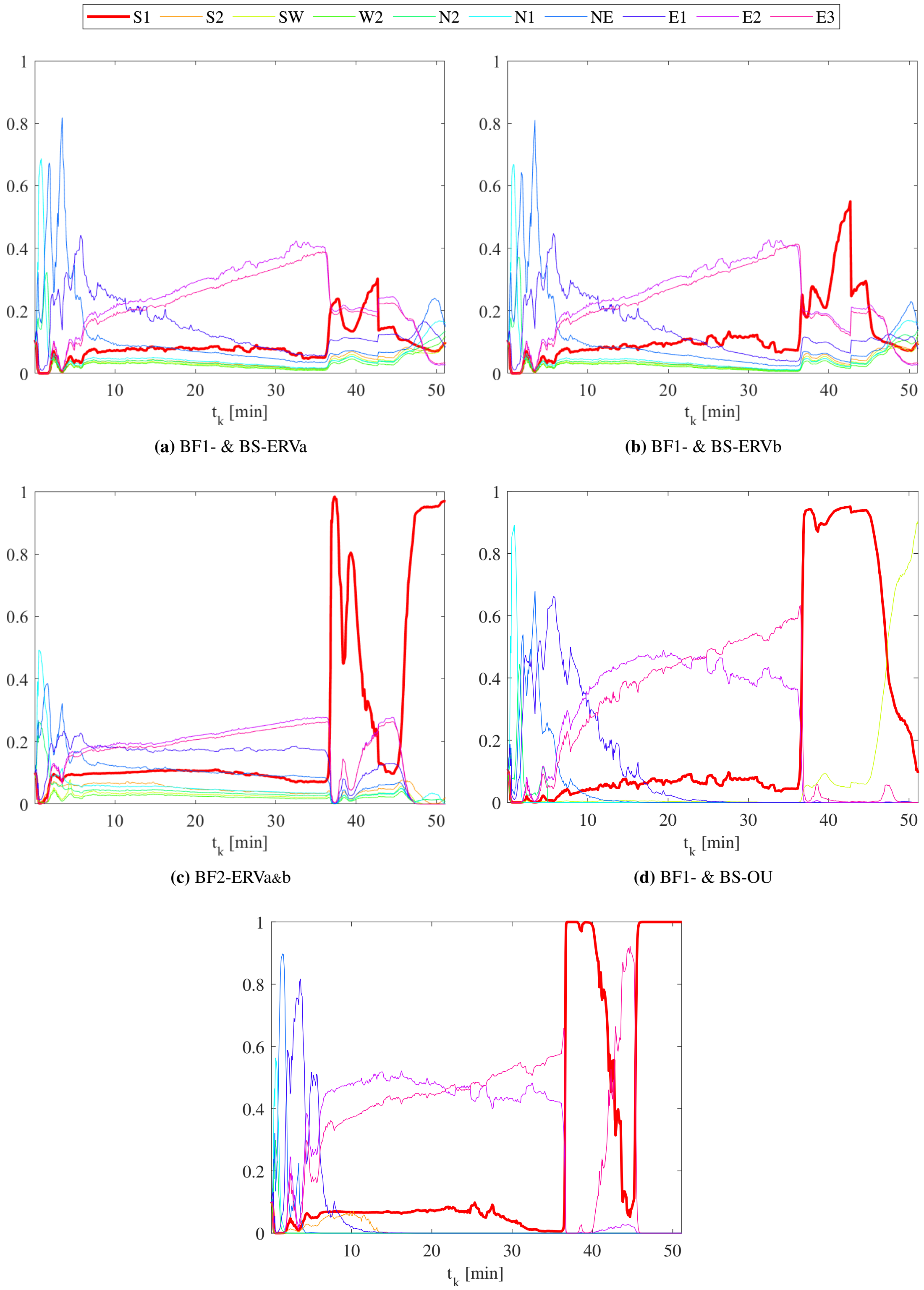


Figure 6.5: Measured trajectory of *Kong Harald*, a passenger ship, travelling from W1 to S1.

The computed destination weights are shown in fig. 6.2. It shows that for the first ~7 minutes, the destination estimates go towards *N1*, *NE* and then *E1*, which are unexpected destinations when looking at the heading direction. A possible explanation for this behaviour is that during the first



(a) BF1- & BS-ERVa

(b) BF1- & BS-ERVb

(c) BF2-ERVa&b

(d) BF1- & BS-OU

(e) BF2-OU

Figure 6.6: Computed destination weights (*y-axis*) at each time step, t_k, over the travelling time for the ship travelling from *W1* to *S1*. The destination weights for the true destination is illustrated as a bold line.

few measurements the model has not developed a good arrival time distribution and can choose destinations that appear random. This initial progress was observed to generally take somewhere between 20 and 50 measurements. For *W1*-*S1* this was about 5 minutes after the initial measurement, then the weights on *E2* and *E3* began to increase. This increment was observed for all cases until the ship changed its heading towards *S1*. Moreover, if a straight line were to be drawn along the initial heading, it would first go through the north-part of the *S1*-ellipse and then continue towards *E3*, closely passing south of its ellipse. Even though the ship is heading inside the *S1*-ellipse, it is not heading towards its centre. Furthermore, the model seemed to favour destinations that are positioned further away if the ship is not heading towards the centre of a destination that is closer to the current position.

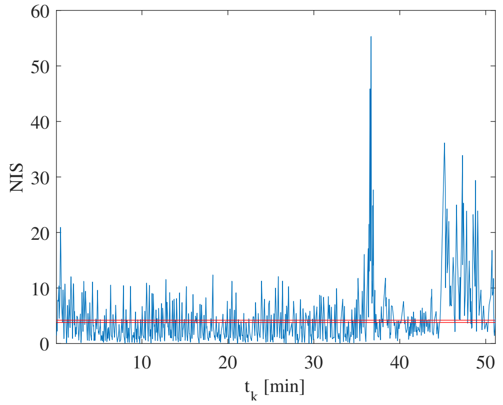
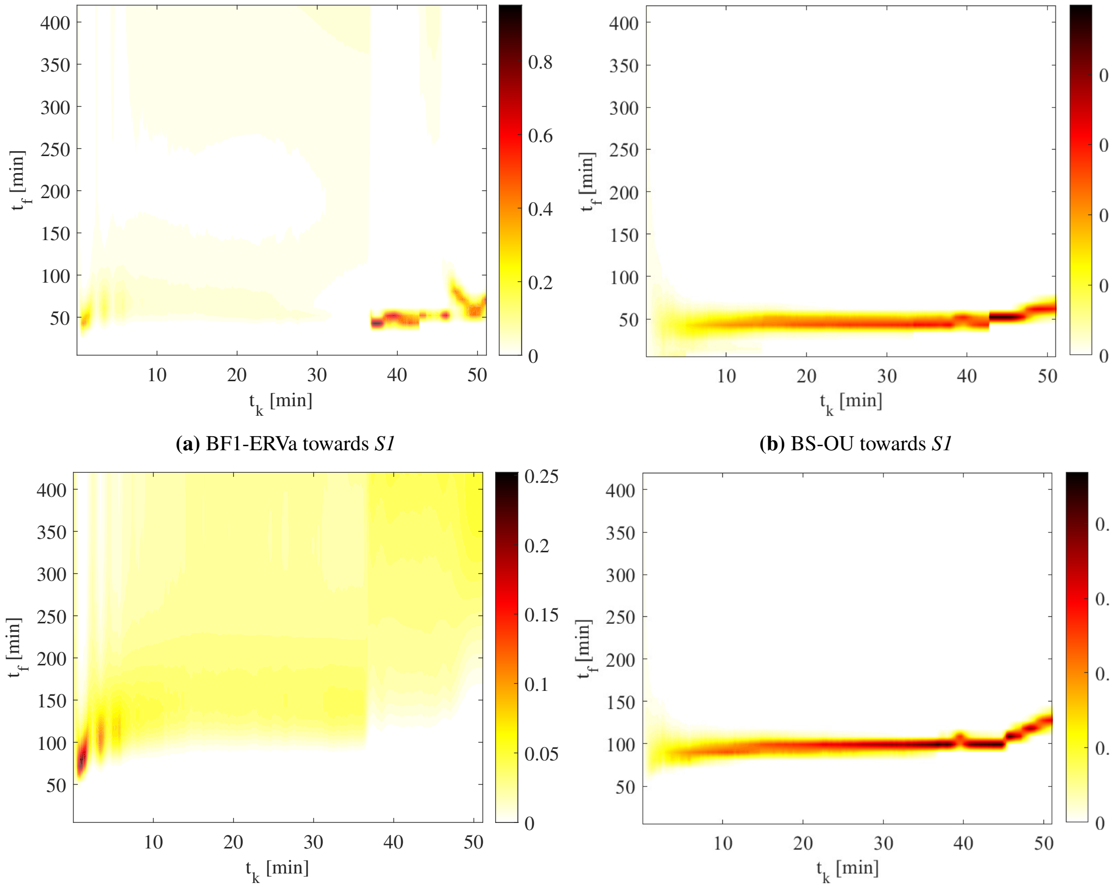


Figure 6.7: Computed NIS at each time step for *W1*-*S1* using BF1-ERVa. The red lines represent the desired values of the average NIS.



(a) BF1-ERVa towards *S1*

(b) BS-OU towards *S1*

(c) BF1-ERVa towards *E2*

(d) BS-OU towards *E2*

Figure 6.8: Arrival time (t_f) probability distribution development over the travelling time (t_k) for the ship travelling from *W1* to *S1*. Computed for the possible destinations *S1* and *E2*, using the model cases BF1-ERVa and BS-OU. The values on the bar going from light to dark represent the probability values.

After about 37 minutes the ship began to change its heading towards *S1*, at that time, much like with the previous scenario, the computed NIS spiked and consequently allowed the model to become overconfident in its estimates. Even though this led to correct estimations, this might not be a desirable trait since it is possible that a wrong destination could be chosen for similar reason. All cases using BF2, and all cases using OU, were substantially affected by this, where almost 100% of the weight was assigned to *S1* during the heading change. An example of the computed NIS for BF1-ERVa at each time step is illustrated in fig. 6.7.

Considering the last few minutes, the ship passed the *S1*-centre after about 46 minutes and began to slow down while continuing east for a few minutes. This can be seen in fig. 6.6 (for most cases), where the *S1* weights drop and the model starts to consider other destinations again.

On the subject of the arrival time distribution, the distribution development for two cases and two destinations are illustrated in fig. 6.8. It shows that BS-OU was more consistent in its arrival time estimations in addition to having narrower distribution compared to BF1-ERVa. BS-OU had a somewhat expected arrival time estimations, where the time it takes to travel to *E2* is approximately double the time it takes to travel to *S1*.

6.2.3 *W1-E2*: Destination inference for a ship travelling from *W1* to *E2*

The ship travelling from *W1* to *E2*, with trajectory shown in fig. 6.9, is a cargo ship which travelled with almost constant velocity the duration of the travel.

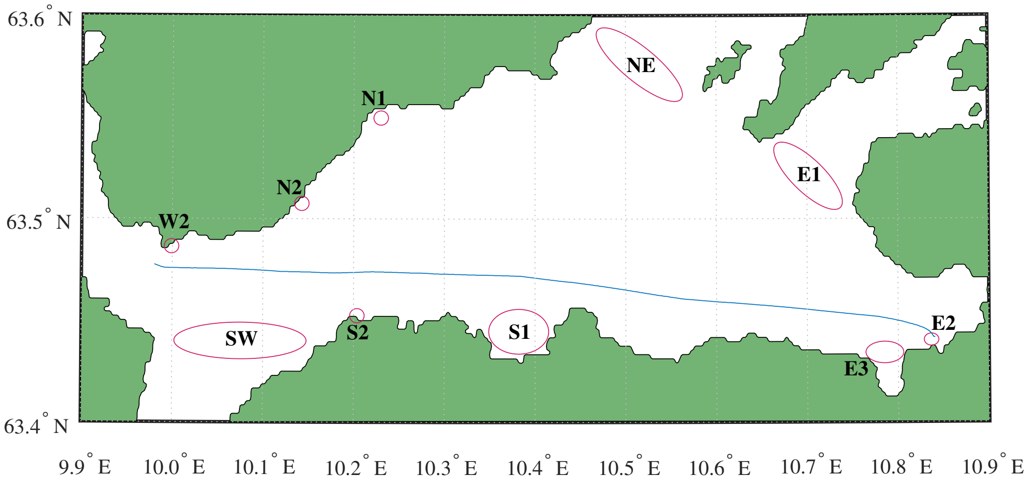
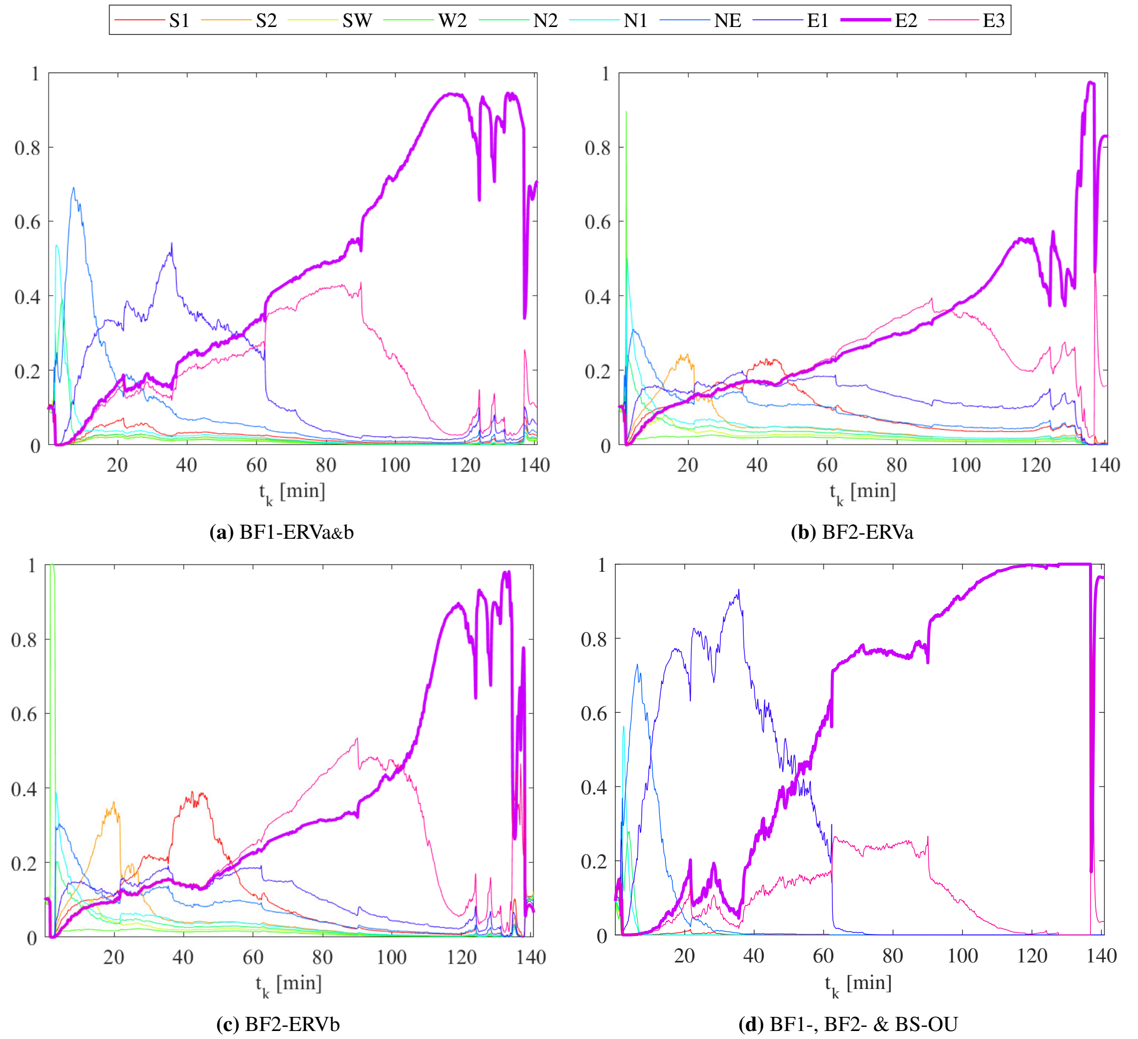


Figure 6.9: Measured trajectory of *Vestland*, a cargo ship, travelling from W1 to E2.

The destination weights development are shown in fig. 6.10. For the most cases the model estimated the ship was sailing towards *NE*, then *E1*, and for the rest of the travel, *E2*. There were no significant differences between the assigned destination weights of the OU models (fig. 6.10d). There was, however, observed a slightly different trend for BF2-ERVa and ERVb contrary to previous scenarios, shown in figs. 6.10b and 6.10c. As the ship got closer to *S2* and then *S1*, their assigned weights increase before then decreasing again after passing them. This effect was more abundant when using ERVb compared with ERVa.

Similarly to the above scenarios, when using OU the model became very confident in its estimates and chose destinations that were straight ahead to be the most likely destination.



(a) BF1-ERVa&b

(b) BF2-ERVa

(c) BF2-ERVb

(d) BF1-, BF2- & BS-OU

Figure 6.10: Computed destination weights (*y-axis*) at each time step, t_k, over the travelling time for the ship travelling from *W1* to *E2*. The destination weights for the true destination is illustrated as a bold line.

6.2.4 *N2-S2*: Destination inference for a ship travelling from *N2* to *S2*

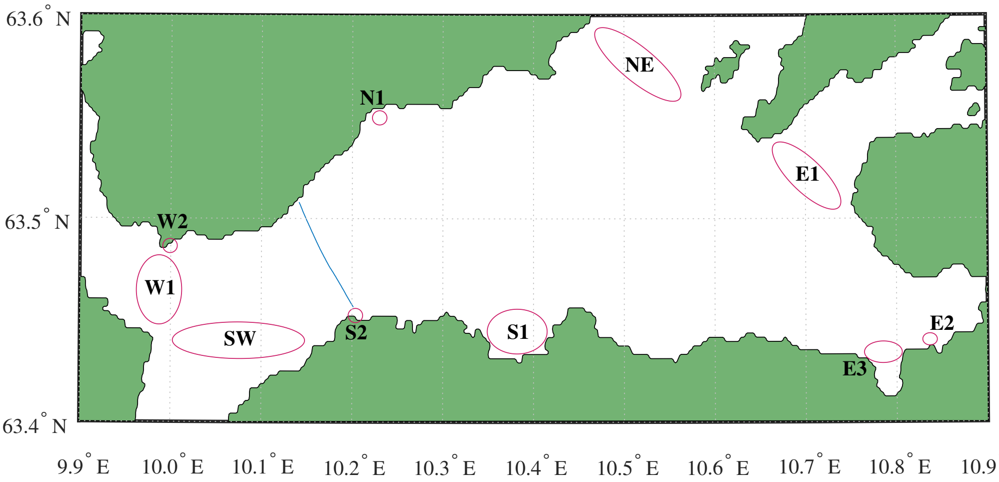
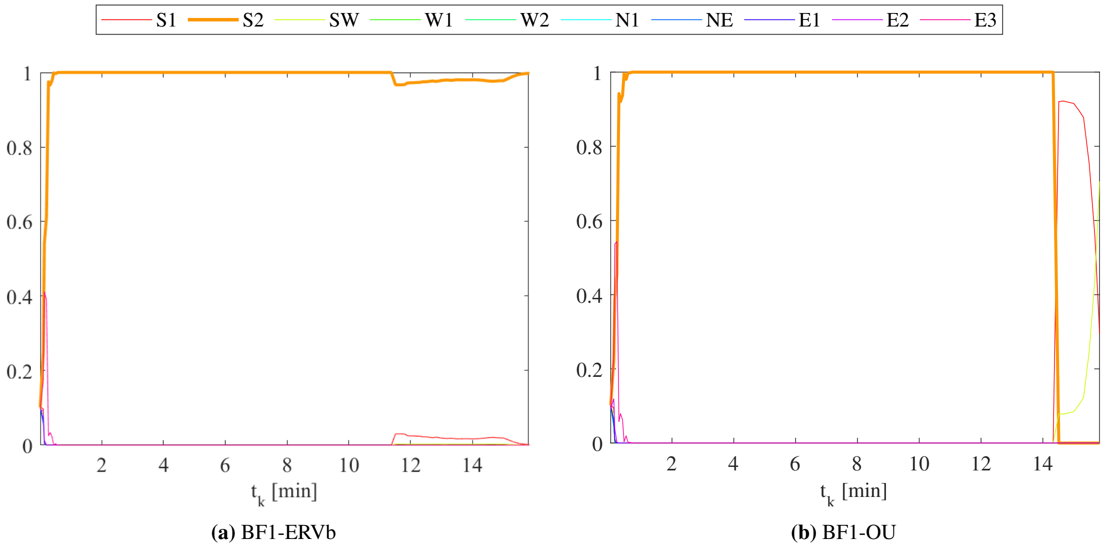


Figure 6.11: Measured trajectory of *Lagatun*, a passenger ship, travelling from N2 to S2.

The ship travelling from *N2* to *S2* is a passenger ship, a ferry, and its trajectory is illustrated in fig. 6.11. This route was included as one of the scenarios to test if a destination could be effectively estimated for a relative short distance with few measurements. All cases estimated the correct destination immediately and were almost 100% certain the duration of the travel. This does not come to as a surprise since the boat sailed in an almost straight line towards its destination. Examples of the destination weights are shown in fig. 6.12 .



(a) BF1-ERVb

(b) BF1-OU

Figure 6.12: Computed destination weights (*y-axis*) at each time step, t_k, over the travelling time for the ship travelling from *N2* to *S2*. The true destination is illustrated as a bold line.

6.2.5 *NE-W1*: Destination inference for a ship travelling from *NE* to *W1*

The trajectory of a cargo ship travelling from *NE* to *W1* is shown in fig. 6.13. The destination weights are shown in fig. 6.15. BF1 and BS had the same result both with ERVa and ERVb. Its weights are illustrated in fig. 6.15a, which shows how the model usually chooses the destination

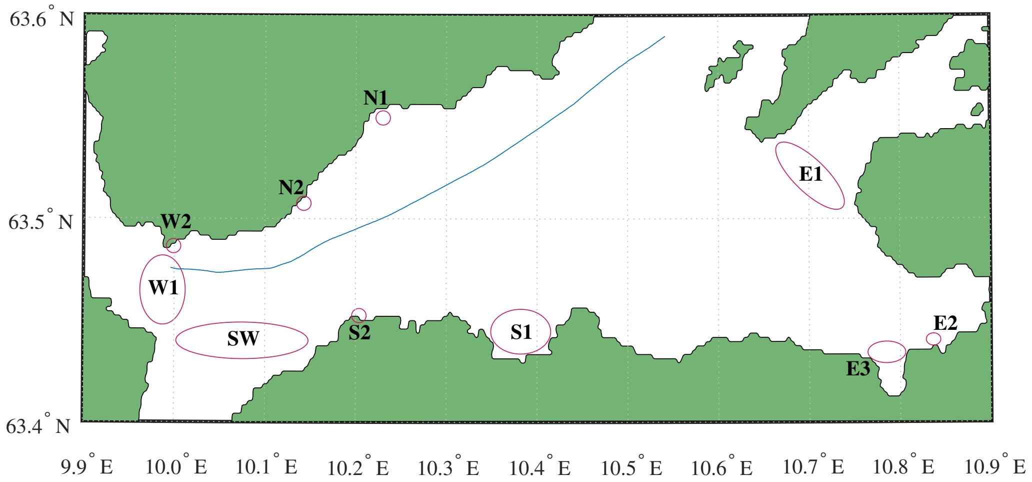
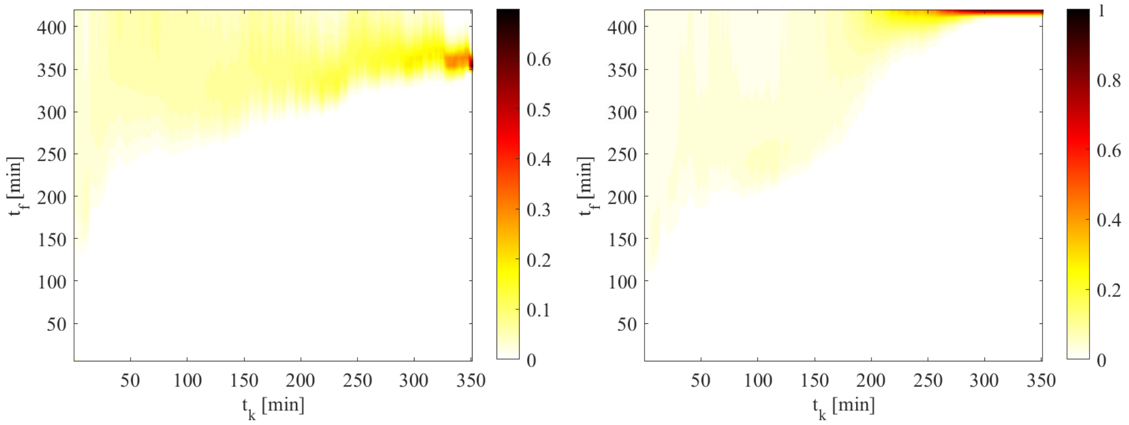


Figure 6.13: Measured trajectory of *Ro Chief*, a cargo ship, travelling from NE to W1.

that is straight ahead but changes the weights when the heading changes slightly. A similar trend was observed for BF2-ERVa (and BF2-ERVb) but with lower destination weights. The assigned weights for these cases were rather low for the first ∼60 minutes. Moreover, using OU resulted in high destination weights that were in the direction of the heading. The BF1-OU and BS-OU performed better compared with BF2-OU in regard of estimating the correct destination earlier.

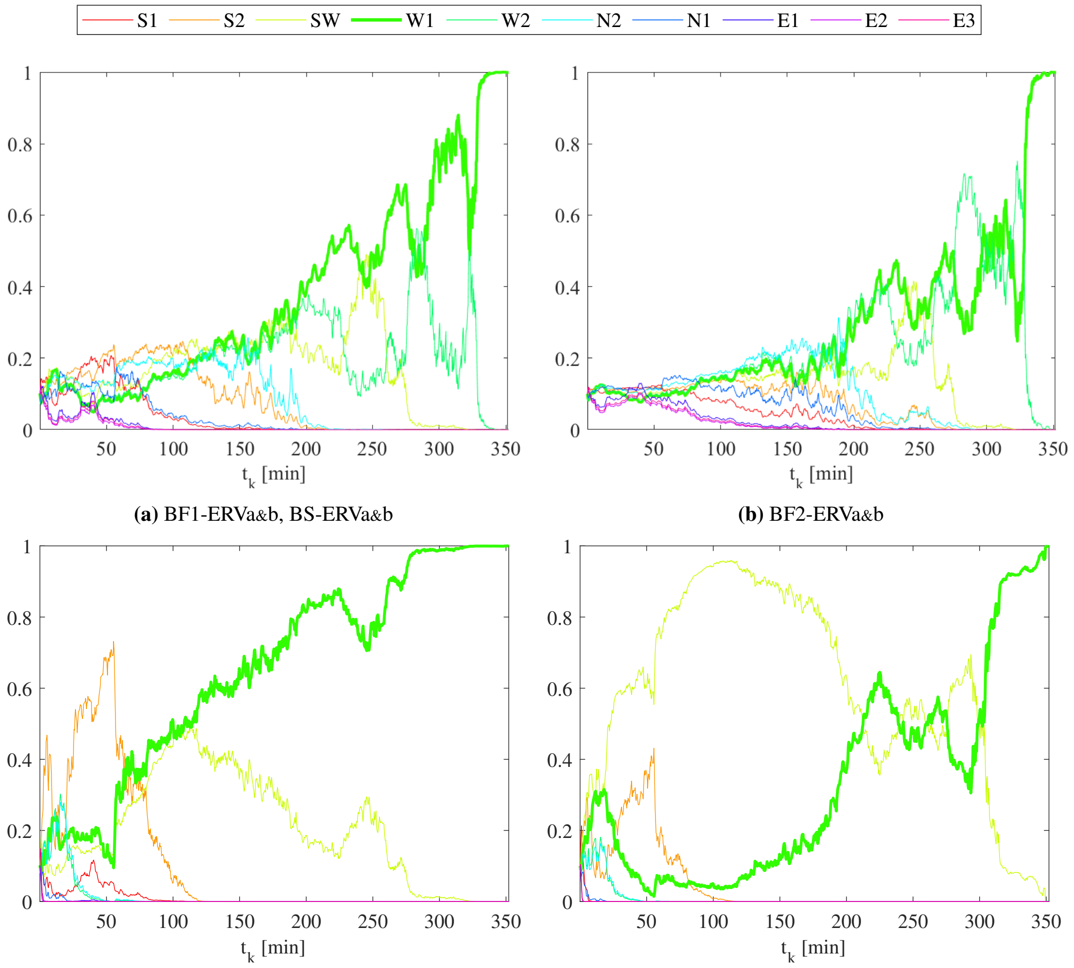
Consider the arrival time distributions, two examples are illustrated in fig. 6.14. All cases had relatively good arrival time estimations for *W1* (the true destination) with its confidence increasing over time, fig. 6.14a shows the distribution for BF1-ERVa. To illustrate how the model handles destinations that the tracked object is moving away from, the arrival time distribution for *S2* is shown in fig. 6.14b. When the tracked object has passed a destination, or simply when it is travelling away from it, the arrival time for the said destination will be estimated to be the highest possible arrival time available. This follows from the mathematics, where the model would continue to increase the arrival time estimates if the arrival time interval considered would go towards infinity. However, in this thesis the highest arrival time considered is 7 hrs; or 420 min.



(a) BF1-ERVa towards *W1*

(b) BF1-ERVa towards *S2*

Figure 6.14: Arrival time (t_f) probability distribution development over the travelling time (t_k) for the ship travelling from *NE* to *W1*. Computed for the destination *W1*, using the model case BF1-ERVa. The values on the bar going from light to dark represent the probability values.



(a) BF1-ERVa&b, BS-ERVa&b

(b) BF2-ERVa&b

(c) BF1- & BS-OU

(d) BF2-OU

Figure 6.15: Computed destination weights (*y-axis*) at each time step, t_k, over the travelling time for the ship travelling from *NE* to *W1*. The destination weights for the destination weights for the true destination is illustrated as a bold line.

6.2.6 The difference between using the alternative mean and the given mean

As mentioned above, the presented results in the previous sections are computed with the alternative mean $\mu^{alt.}$. The alternative mean and the given mean gave the same results for all model cases with one exception, namely BS-OU. Using the smoothing approach with the mean given in [6], along with the OU motion model, the destination inference computation produced peculiar results. For the short travel of *N2-S2* the two mean variations produced the same results. However, only one of the other four scenarios produced a reasonable destination weights using the given mean, namely *NE-S1*, while for the remaining scenarios, the results were far from expected. The different results of the two variations are illustrated in fig. 6.16 for *NE-S1*, and another example, of *W1-E2*, is illustrated in fig. 6.17, showing how the given mean produced very different results. A comparison for all scenarios is given in Appendix A.II.

Consider the motion models, the major difference between them is the added vector, **m**; the general tracking model is shown again in eq. (6.1) for convenience. Where, for the ERV model

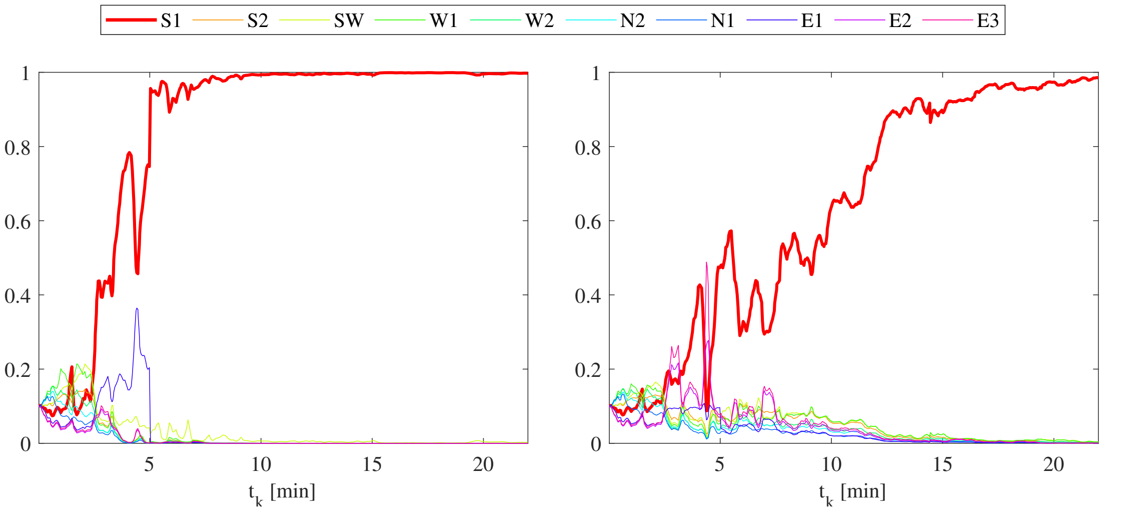


Figure 6.16: Destination weights over time for *NE-S1* using BS-OU with, (*left:*) the alternative mean $\mu^{alt.}$ and, (*right:*) the mean given in the article.

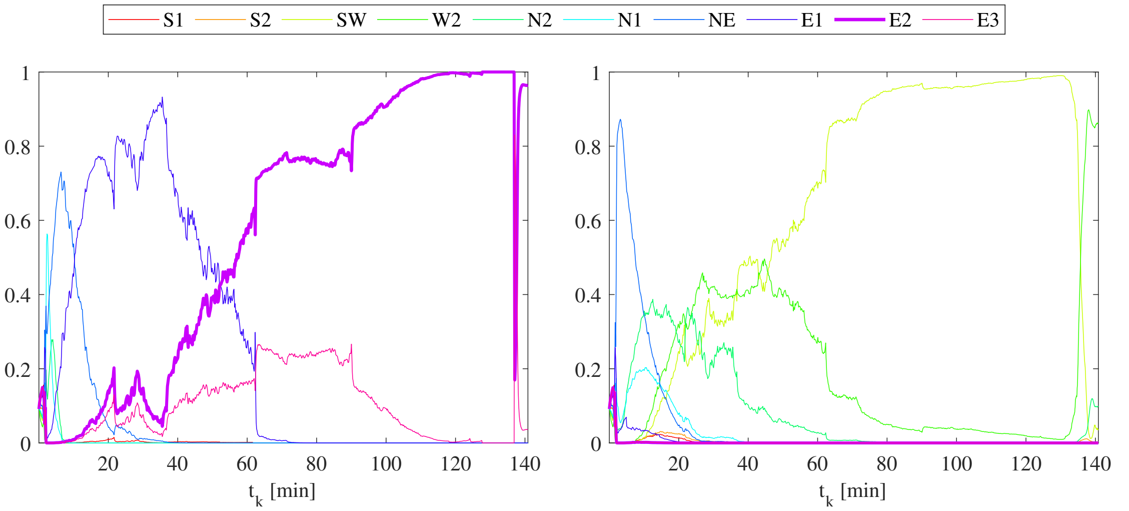


Figure 6.17: Destination weights over time for *W1-E2* using BS-OU with, (*left:*) the alternative mean $\mu^{alt.}$ and, (*right:*) the mean given in the article.

with zero drag and zero reversion strength, **m** becomes small, and if the final velocity is assumed to be zero, **m** also becomes zero. On the other hand, for the OU model, **m** will never be zero as long as the mean velocity is nonzero. Hence, it seems as the added vector **m** is the determining factor of whether using the given mean will result in meaningful destination weights or not.

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{m} + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathscr{N}(\mathbf{0}, \mathbf{Q}) \tag{6.1}$$

Now, for the bridging models presented in [6] (BF1 and BS), the product identity in theorem 1 was used for the derivations opposed to theorem 2. It is possible that due to the complexity of the probability distributions, theorem 1 is not suited for the derivations, and a small portion of information might be lost. Additionally, since the smoothing equations has an added layer of equations, and by using equations with missing information many times will increase the inaccuracy. Hence, it will affect the smoothing equations more compared to the filtering equations.

To test the hypothesis that the added vector **m** is the determining factor of the performances of BS, opposed to BS-OU simply being a bad combination. The BS-ERVa model case was run with drag and reversion coefficients resulting in a nonzero **m**. Note that this is not a good representation of the ship's motion that is travelling with a relatively constant velocity. Figure 6.18 illustrates the computed destination weights for *W1-E2* using BS-ERVa with drag and reversion coefficients. It shows how the two variations no longer produce the same results, where using the alternative mean gave similar result as BS-ERVa without drag and reversion coefficients, but using the given mean had very different results. However, not as severe as for the BS-OU case.

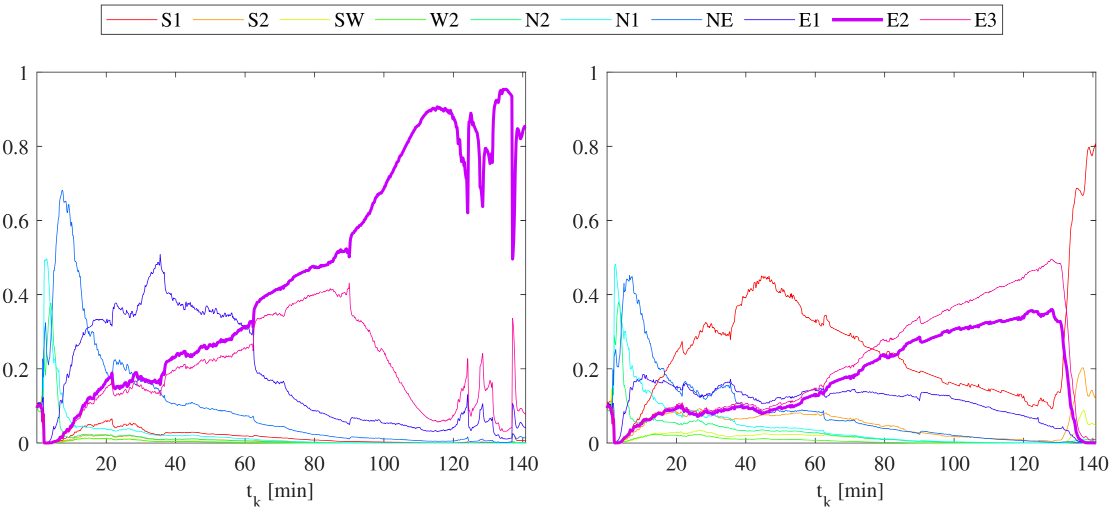
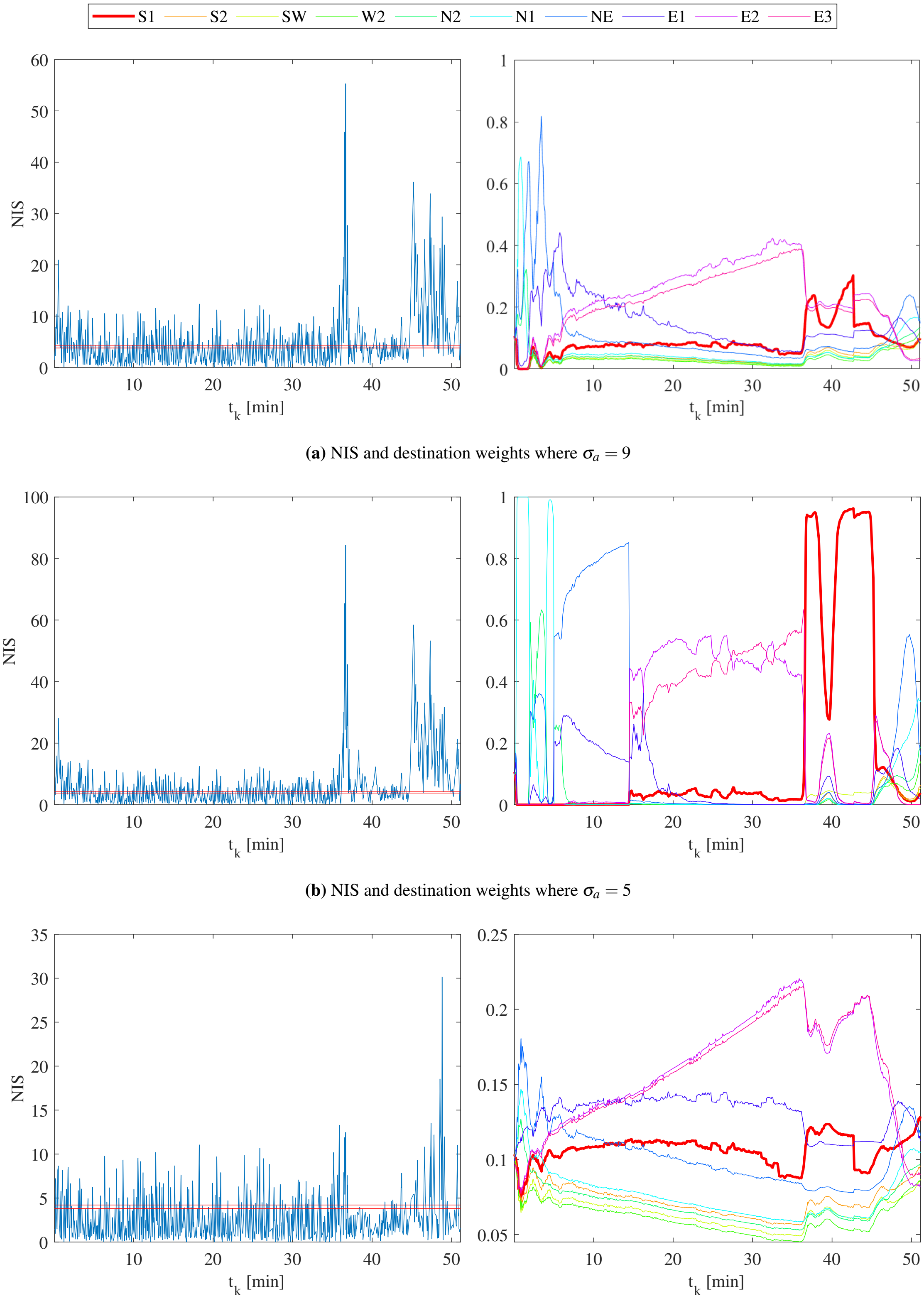


Figure 6.18: Destination weights over time for *W1-E2* using BS-ERVa including reversion strength and drag force. (*Left:*) the alternative mean $\mu^{alt.}$ and, (*right:*) the mean given in the article.

6.2.7 The importance of correctly tuned model

As mentioned earlier, the computed destination weights are highly affected by the NIS. With too high NIS values, the model will get overconfident and will assign destinations with large weights. Whereas, too low NIS values will let the model be too indecisive and the assigned destination weights will change very slowly over time and remain similar for multiple destinations. To illustrate the affect of poorly tuned covariance, the process variance, σ_a, was varied for *W1-S2* using BF1-ERVa case model, while keeping all other parameters constant (given in table 5.3). The computed NIS at each iteration, along with the corresponding destination weights, are illustrated in fig. 6.19a, where the lower and upper NIS limits are represented as red lines on the NIS graph.[1] It is desirable to keep the average NIS between those limits. As shown in fig. 6.19a, when the ship was travelling at a nearly constant velocity ($t_k \in [0,35]$ min), the NIS values were fluctuating around the desired values; perhaps slightly below. Then, as mentioned above, the ship changed its heading and consequently the NIS values spiked. When σ_a was reduced this spike was amplified and the destination weights were much higher, this is shown in fig. 6.19b.Figure 6.19c shows the results from increasing the value of σ_a, it shows how the NIS spike was reduced greatly. However, the average NIS was pushed far below the desired values, and consequently, the assigned destination weights had little change over time and was kept around 0.1 for most destinations.

[1] *Remark:* the NIS is relatively noisy, this could have been reduced to some extent with better tuning, but due to the time frame of this thesis, this was not considered a significant factor for its purpose.



(a) NIS and destination weights where $\sigma_a = 9$

(b) NIS and destination weights where $\sigma_a = 5$

(c) NIS and destination weights where $\sigma_a = 20$

Figure 6.19: (*Left:*) computed NIS at each time step and, (*right:*) computed destination weights for *W1-S1* using BF1-ERVa. Two red lines on the NIS graphs mark the upper and lower limits, where it is desirable to keep the average NIS between.

6.2.8 Discussion, destination inference

In the tested scenarios with the model cases shown in table 6.1, a trend of how the different models behaved was observed. It did not make a significant difference if BF1 or BS was used since these produced the same, or nearly the same, destination weights when using the alternative mean $\mu^{alt.}$. However, when using the mean given in [6] for BS, where the tracking model had a nonzero $\mathbf{m}$, the resulting destination weights were poor, or unacceptable in some scenarios. For BF1, on the other hand, it did not matter if the given mean or the alternative mean was used, applicable for both nonzero and zero valued $\mathbf{m}$. Therefore, it would be preferable to use BF1 over BS for tracking maritime vessels since it is not only less complex, it is also not affected by the use of either the alternative or the given mean.

Often when there is a change in the heading angle, a sudden increase in the NIS values was observed. This leads to the model becoming overconfident and it will assign high weights for certain destinations. At these events it was observed that BF2 was more subjected to this change compared with BF1, where it had more severe jumps in the destination weights. Furthermore, when using the OU motion model, the model also became overconfident, not only when NIS was high but rather over the whole travel. This is not necessarily a bad thing as the model seemingly had a decent destination estimates. This could be due to the OU model considers the previous velocities as one of the variables in its destination estimates. However, this relatively good estimation was usually true when combined with BF1 (BF1-OU) or BS, but not as good with BF2 (BF2-OU).

The difference between ERVa and ERVb was often insignificant, where both produced were similar trends with slightly different values. Hence, it does not have a significant affect to including the course angle as one of the state variables.

6.3 Future predictions

Using the model cases in table 6.1 a future trajectory for each destination can be predicted, where the predicted trajectory should in theory revert towards its destination because of its bridging property. A few examples will be presented in this section, and if the reader is interested more are provided in Appendix A.III.

Regarding the future prediction illustrations presented below. The true measured trajectory is represented by a blue line, on which there are two other symbols, a blue circle and a blue star. The blue circle represents the position at the current time, t_k, and the blue star represents the measured position t_* minutes after the current time, i.e. at time t_{k+*}. The predicted trajectory for a tracked object travelling towards the destination MAP estimate (the highest destination weight at t_k), is illustrated with a red line going from the current position to the predicted position at time t_{k+*}, marked by a red star. The area of uncertainty, the covariance, is illustrated as a grey area around the predicted trajectory. Additionally, all destinations that have assigned weights over 0.1 at the current position will be considered. The predicted positions at time t_{k+*} for these are illustrated with a red dot. Their trajectories are, however, not drawn on the map, but rather only the area of uncertainty. This was done for easier distinction between the destination MAP estimate and the other considered destinations. The symbols are summarised in table 6.2.

Table 6.2: Legends for future prediction illustrations

Symbol	Description
○	The current position at time t_k from which future trajectories are computed from
—	True measured trajectory
⋆	True position t_* minutes after the current time
⋆	Predicted position t_* minutes after the current time, for the destination MAP estimate
—	Predicted trajectory for the destination MAP estimate
●	Predicted position t_* minutes after the current time, for destination with weights over 0.1
●	Area of covariance

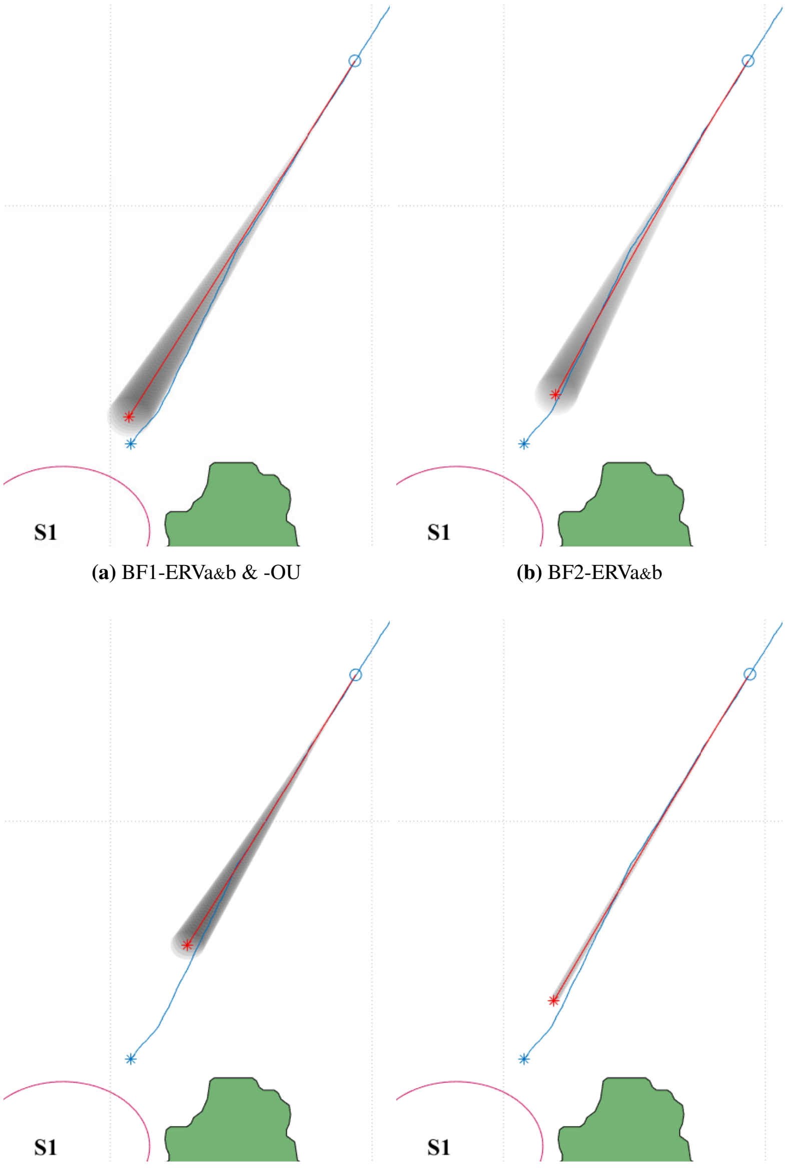
6.3.1 *NE-S1*: Future predictions for a boat travelling from *NE* to *S1*

For the *NE-S1* scenario, the destination MAP estimate was *S1* for all cases for the majority of the travel time, but with different weights. Therefore, only *S1* is considered here for future predictions. Figure 6.20 illustrates the predicted position and trajectory 10 min after the current time, $t_k = 10$ min.

The effect of using ERVb instead of ERVa was insignificant, as these predicted almost identical future states. In fact, this was the case for all scenarios tested. Furthermore, when paired with BF1, OU motion model resulted in almost the same predictions as BF1-ERVa&b. BF1 had a decent predictions up to about 5 min, but started to slow down after that as it got nearer its destination. The model assumed the boat would start to slow down earlier than in reality. BF2-ERVa&b had similar results but where the velocity started to decrease both earlier and faster (higher negative acceleration). Similar behaviour was observed with BF2-OU, but much more severe, where the predicted velocity was much lower than in reality. Lastly, BS had similar trend as BF2-ERVa&b, where it started to slow down early as it approached the destination. Additionally, its covariance was kept very small at all times, meaning that the model was highly overconfident in its predictions, which is very unrealistic.

6.3.2 *W1-S1*: Future predictions for a ship travelling from *W1* to *S1*

For the remaining examples, all destinations that had assigned weight, at t_k, higher than 0.1 (the average) were considered. The predicted position t_* minutes after t_k for the destination MAP estimate is represented by a red star, whereas the position for the other considered destinations are represented by a red dot. Additionally, only the trajectory for the destination MAP estimate has its trajectory drawn (a red line). Note that the area of uncertainty is illustrated as a grey transparent area, and when multiple such areas are drawn on top of each other they appear darker.



(a) BF1-ERVa&b & -OU

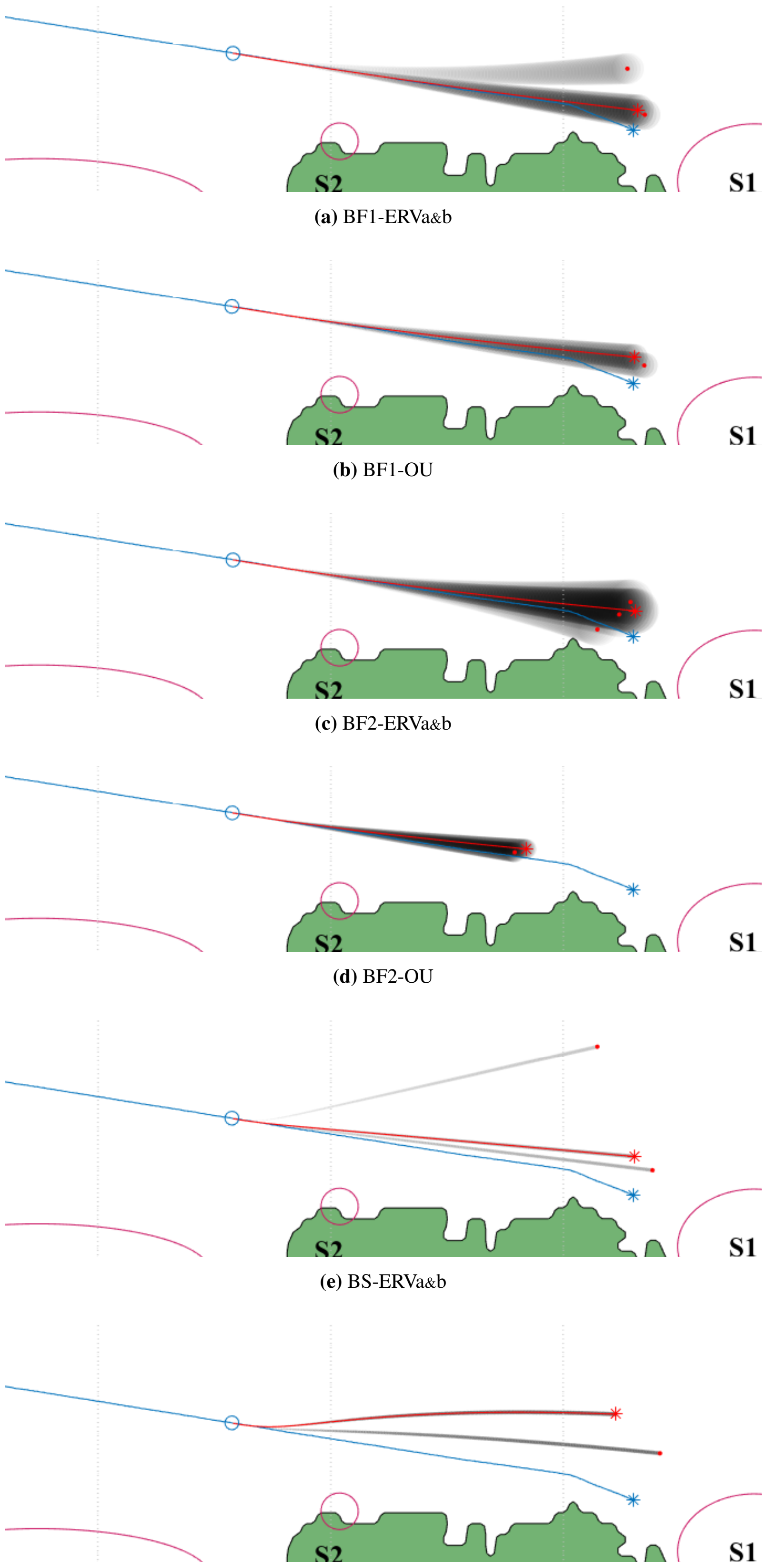
(b) BF2-ERVa&b

(c) BF2-OU

(d) BS-ERVa&b & -OU

Figure 6.20: Predicted future trajectory from $t_k = 10$ min (blue circle), $t_* = 10$ min forward in time for the scenario *NE-S1*. The symbols are described in table 6.2.

Figure 6.21 illustrates the predicted trajectories from the current time $t_k = 20$ min towards $t_{k+*} = 40$ min (20 min forward in time). The destination MAP estimate for all model cases was either *E2* or *E3* at the current time, and only BF2-ERVa&b assigned *S1* (the true destination) with a weight higher than 0.1. BF1-ERVa&b and BF1-OU had very similar predictions, with the exception of BF1-ERVa&b considering one more destination, i.e. very good predictions while the velocity and heading remained nearly constant considering the long-term prediction of 20 min. BF2-ERVa&b also had a similar results for the destinations further away, where the predicted travelling distance was similar to the true distance travelled after the same amount of time. However, for the destination *S1* the predicted velocity began to decrease early on, which follows from the ship getting near the destination.



(a) BF1-ERVa&b

(b) BF1-OU

(c) BF2-ERVa&b

(d) BF2-OU

(e) BS-ERVa&b

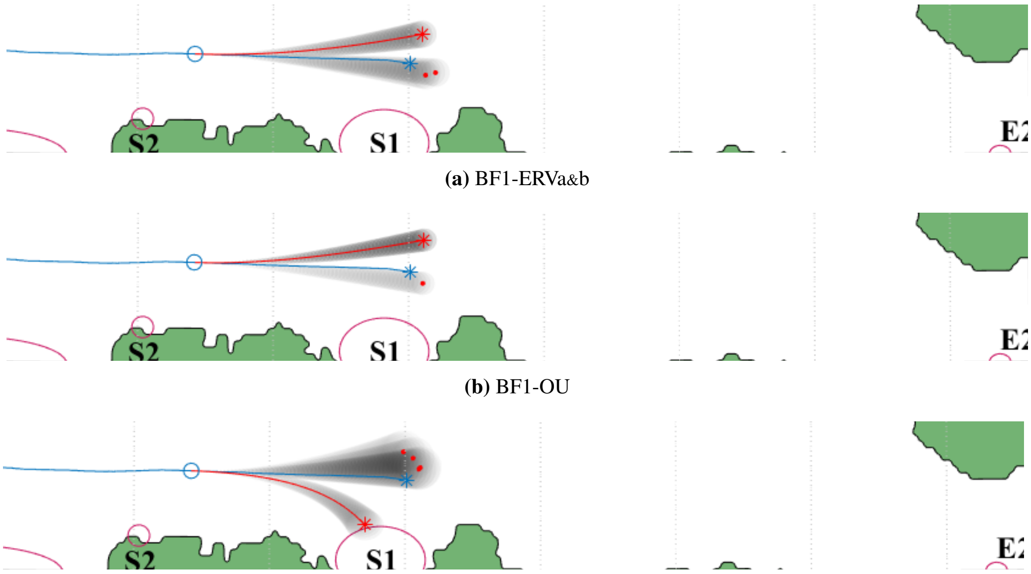
(f) BS-OU

Figure 6.21: Predicted future trajectory from $t_k = 20$ min (blue circle), $t_* = 20$ min forward in time for the scenario *W1-S1*. All destinations with weights over 0.1 at t_k are considered. The symbols are described in table 6.2.

Furthermore, the same trend was observed for BF2-OU and the BS model cases as for the previous scenario. BF2-OU predicted very low velocities at all times, and BS had a very small area of uncertainty. Additionally for BS, when the considered destination was not in a straight line of the current heading, the trajectory had the following trends. (1) For BS-ERVa&b the ship was predicted to have a constant turn rate until its heading was towards its destination. (2) For BS-OU the ship was predicted to change its trajectory towards a position where it could continue in a straight line towards its destination while remaining the same heading as was at t_k. Since the resulting predictions were this poor for these models, the predicting trajectories will not be presented in the following sections. However, they are given in Appendix A.III

6.3.3 *W1-E2*: Future predictions for a ship travelling from *W1* to *E2*

For the scenario *W1-E2*, predictions further into the future, compared with previous examples, were evaluated. Examples of the predicted positions and trajectories are illustrated in figs. 6.22 and 6.23. Figure 6.22 shows the predicted trajectories computed from the current time $t_k = 40$ min, predicting 25 min forward in time, and fig. 6.23 shows the computed trajectories from the current time $t_k = 70$ min, predicting 40 min forward in time. At $t_k = 40$ min, the estimated destinations were still far away, and the velocity decrement observed for the previous examples, was not observed here. However, the contrary was observed, where the predicted velocity increased slightly before then decreasing again as it started to approach the destination. As shown in fig. 6.23, BF2-ERVa&b began to decrease the velocity very early, and both BF1-ERVa&b and BF1-OU behaved very similarly. However, as the prediction was expanded further in time, their predicted states started to slightly deviate from each other, here BF1-OU restricted its velocity to be kept closer to a constant value.

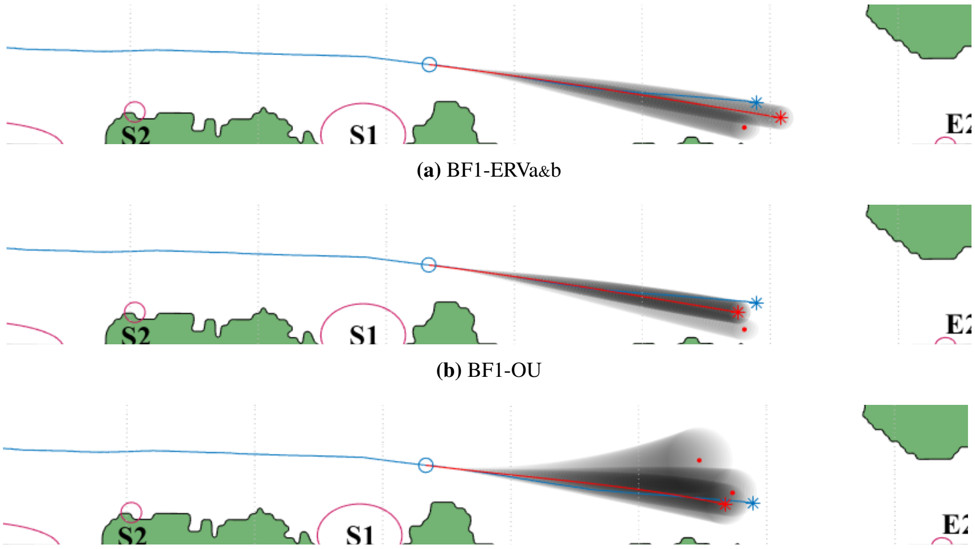


(a) BF1-ERVa&b

(b) BF1-OU

(c) BF2-ERVa&b

Figure 6.22: Predicted future trajectory from $t_k = 40$ min (blue circle), $t_* = 25$ min forward in time for the scenario *W1-E2* (*E2* is positioned in the south-east corner, slightly out of frame). All destinations with weights over 0.1 at t_k are considered. The symbols are described in table 6.2.



(a) BF1-ERVa&b

(b) BF1-OU

(c) BF2-ERVa&b

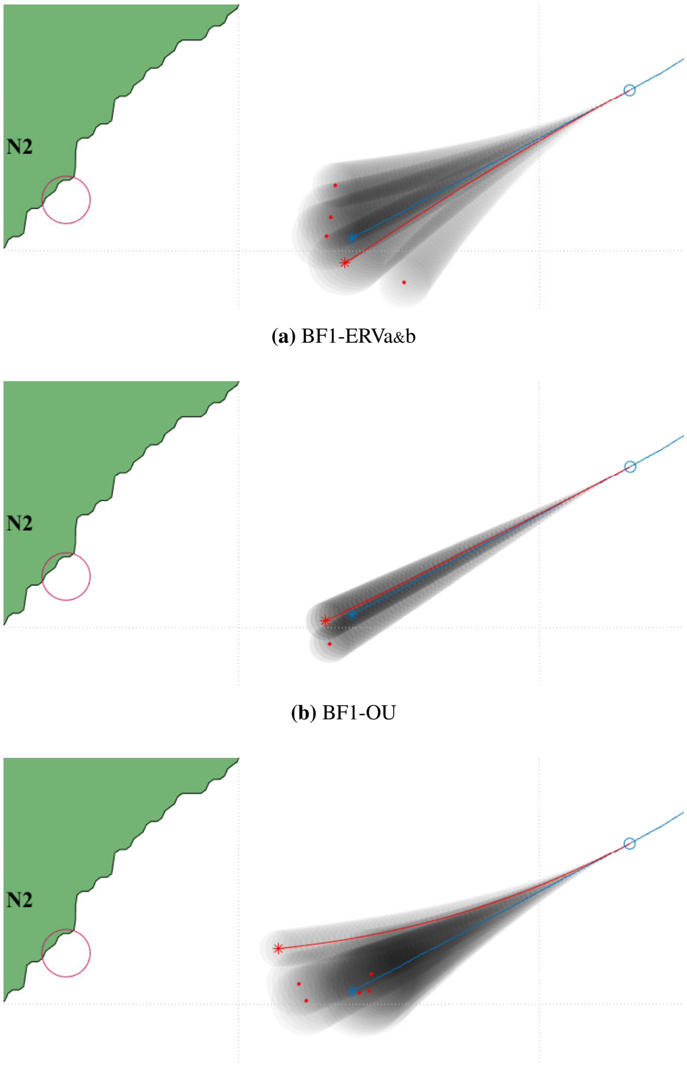
Figure 6.23: Predicted future trajectory from $t_k = 70$ min (blue circle), $t_* = 40$ min forward in time for the scenario *W1-E2* (*E2* is positioned in the south-east corner, slightly out of frame). All destinations with weights over 0.1 at t_k are considered. The symbols are described in table 6.2.

6.3.4 *NE-W1*: Future predictions for a ship travelling from *NE* to *W1*

The scenario *NE-W1* had a rather indecisive destination weights for the most model cases, where multiple destinations had assigned weights over 0.1. Figure 6.24 shows an example of how the model will predict the future trajectories for multiple destinations that are in different directions. The positions and trajectories are computed from $t_k = 125$ min, and ending at $t_{k+*} = 185$ (1 hour after the current time). For the most part, the model predicted the distance travelled after 60 min to be substantially further than for the true measurements. With the model cases BF1-ERVa&b and -OU the predictions were decent up to about $t_* = 30$ min into the future, but started to slowly deviate from the true measurements after that. On the other hand, for BF2-ERVa&b the predictions were decent up to about $t_* = 25$ min, but after that it started to increase the velocity. In some situations, this increase in velocity is related to too low arrival time estimates, and the model has a desire to rush towards a destination. But in others, this seems to come from the fact that the destination is simply far away.

6.3.5 Discussion, future predictions

The model cases shown in table 6.1 were not only used for destination inference but for future predictions as well. There was no significant difference in using the alternative mean and the mean given in [6] when computing the predictions. How well the model cases performed in predicting future states, varied dramatically between cases. Those that had reasonable predictions for at least 15 min into the future were BF1-ERVa&b, BF1-OU, and BF2-ERVa&b. Depending on the scenario, these models usually had good predictions up to about 15-30 min of future predictions, where the true measured positions landed within the area of uncertainty. After that the predicted positions started to diverge from the measured values.



(a) BF1-ERVa&b

(b) BF1-OU

(c) BF2-ERVa&b

Figure 6.24: Predicted future trajectory from $t_k = 125$ min (blue circle), $t_* = 60$ min forward in time for the scenario *NE-W1*. All destinations with weights over 0.1 at t_k are considered. The symbols are described in table 6.2.

The predicted velocity profile seemed to be mainly determined by two factors, the distance to the destination and the arrival time estimates. The arrival time MAP estimate, computed at time t_k, was used to compute the future predictions. For a case where the arrival time was overestimated, the model predicted lower velocities, whereas, for underestimated arrival time, the model predicted high velocities. Additionally, if the predicted position was substantially far away from the destination a higher velocity was predicted, and as it got nearer the velocity was predicted to decrease. Whereas, in reality, the tracking object travelled with nearly constant velocity for much longer, and only started to slow down a few hundred meters before arriving at its destination. This behaviour was most exaggerated for BF2-ERVa&b, where it started to slow down very early, or about 15-20 km from the destination. Furthermore, The difference between BF1-ERVa&b and BF1-OU was often insignificant with few exceptions.

Concerning the model cases that had poor future predictions. With BF2-OU, the predicted trajectory had a similar form to BF1-OU. However, the predicted velocity started to decrease shortly after the current time, and the it was kept at low values for the whole prediction. For the BS cases, the area of uncertainty was unrealistically small for the predicted trajectory, and when the trajectory was not in a straight line, its form was usually unrealistic as well. Normally, Bayesian smoothing is used to smooth out previous state estimates, for instance RTS smoother starts from the current state and smooths out previous state estimates backward in time. However, the smoothing equations used in this thesis are structure such that the next step predictions estimates the state and smooths it out with respect to the pseudo-measurement in one step. At first the covariance might be adequately large, then, since the model assumes the pseudo-measurement to be the next true measurement, it "corrects" the state estimates via smoothing and becomes more confident, reducing the covariance in the process.

Chapter 7

Conclusion

Three bridging models for predicting a tracked object's intent, presented in [5, 6], have been studied in detail. These are, (1) (BF2) a Bayesian filtering approach bridging the current state with the final destination via a joint state of the current and the final state, (2) (BF1) a Bayesian filtering approach bridging the current state with the final destination via a pseudo-measurement, and (3) (BS) a Bayesian smoothing approach also bridging the current state with the the final destination via a pseudo-measurement.

The mathematical derivation of the two latter models was found to have an alternative method from the one used in the article. These are two different methods, or product identities, for finding the mean and covariance of the product of two Gaussian distributions. Using these two methods for the derivations resulted in two slightly different equations for the mean, while the covariance remained the same for both methods. The derivation of the filtering approach (BF1) uses the product identity once, whereas the smoothing approach (BS) uses it thrice. Consequently, the smoothing mean given in the article differs more from the alternative mean, compared with the filtering mean. The two variations were compared with respect to how they performed in destination inference and future predictions, using real measurements. The general tracking model used in this thesis includes an added vector that encourages the model to revert towards a desirable value. It was observed that for models where the added vector is zero, the two variations had no significant difference in their computed values. However, for models where the added vector is nonzero, the smoothing approach resulted in peculiar destination estimates when using the article's mean, whereas using the alternative mean resulted in more reasonable destination estimates. Moreover, only the smoothing approach was affected by this, and the filtering approach resulted in the same destination estimates for both means. To conclude, the product identity used in the article is not suited for smoothing derivations, and it is possible that some information might be lost when using for the probability distributions in the article.

The bridging models were tested in how they perform in predicting the intended destination, as well as predicting future states, using AIS measurement of maritime vessels as real time measurement. Two motion models were considered, ERV model and OU model, additionally, two different state vectors were used in the ERV model. One state vector contains the position and their derivatives, where the other contains the course angle in addition to the position. All pairs of a bridging model and a motion model were tested, nine in total.

Considering the destination inference, the computed destination weights are highly affected by the NIS values. For low NIS values the destination weights are kept low, whereas high NIS values lead to high destination weights. This is because for high NIS the model gets overconfident in its estimates. When a vessel was travelling with a constant heading, the NIS remained relatively low, but when the vessel started to turn the NIS values spiked. At these high NIS events the model increased the confident level of a certain destination, this destination often being the one that was either closest to the vessel at that time, or the one in a straight line of the heading. The BF2 approach was most acceptable to this NIS change and had the largest jumps in the destination weights. It did not make much of a difference whether BF1 or BS was used as these resulted in very similar destination weights. Additionally, the same can be said about the two different state vectors. By including the course angle as one of the state variables did not make a significant difference. Furthermore, using the OU model resulted in a very confident model, regardless of the NIS values. However, by pairing the OU model with either BF1 or BS resulted in a better prediction compared with BF2. These two usually had a good intent predictions where it tented to favour destinations that were in a straight line of the heading.

For the future state predictions, the performance of the different models varied with great extent. The smoothing approach is inadequate for the use of predicting future states as it is unrealistically confident. With BF2 the predicted velocity profile was proportional to the distance to the destination, where it started to decrease the velocity very early when it got nearer the destination. The best bridging model for future predictions was BF1. It did not matter if the course angle was included in the state vector or not, since both resulted in the same predictions. Both the ERV model and the OU model produced reasonable predictions, where OU was perhaps slightly better at keeping the predicted velocity constant when getting nearer the destination, where ERV had started to reduce the predicted velocity too as it got nearer the destination.

Chapter 8

Suggestions for future work

The following future work is suggested:

- Use a dynamic variance for the destination inference. In this thesis the tuning of the covariance parameters was made manually and they were kept constant for the duration of the travel. However, a more realistic approach would be to use different variance during different events to be able to keep the NIS values at a more desirable level. This would allow for the intent computations to be less bias. For instance, letting the variance be a dynamic variable that could be determined by the means of an optimisation problem.

- Develop a reworked algorithm that can take into account stationary obstacles for destination inference. As of now the algorithm assumes clear path from its current position towards all possible destinations. A simple solution to this is to divide the considered map area into multiple smaller areas where stationary obstacles mark their margins, e.g. islands, headlands etc.

- Use guidance model to simulate future states. The velocity profile of the predicted future states was not always realistic, especially for long-term predictions over 15 min. The velocity profile was both affected by the distance to a destination and the estimated arrival time. However, the predicted trajectory usually represented a realistic route towards a destination. Therefore, using the trajectory as setpoints, a more realistic prediction could be achieved by simulating the movement of a vessel using a guidance model, e.g. line of sight model.

Bibliography

[1] O. Levander. Autonomous ships on the high seas. *IEEE Spectrum*, 54(2):26–31, 2017.

[2] International Maritime Organization. Guidelines for the approval of alternatives and equivalents as provided for in various imo instruments, 2013. MSC.1/Circ.1445.

[3] E F Brekke, E F Wilthil, B-O H Eriksen, D K M Kufoalor, Ø K Helgesen, I B Hagen, M Breivik, and T A Johansen. The autosea project: developing closed-loop target tracking and collision avoidance systems. *Journal of Physics: Conference Series*, 1357:012020, 2019. DOI: `10.1088/1742-6596/1357/1/012020`. URL: `https://doi.org/10.1088%2F1742-6596%2F1357%2F1%2F012020`.

[4] Simen Hexeberg. *AIS-based Vessel Trajectory Prediction for ASV Collision Avoidance*, Norwegian University of Science and Technology, 2017.

[5] B. I. Ahmad, J. K. Murphy, P. M. Langdon, and S. J. Godsill. Bayesian intent prediction in object tracking using bridging distributions. *IEEE Transactions on Cybernetics*, 48(1):215–227, 2018.

[6] J. Liang, B. I. Ahmad, R. Gan, P. Langdon, R. Hardy, and S. Godsill. On destination prediction based on markov bridging distributions. *IEEE Signal Processing Letters*, 26(11):1663–1667, 2019. DOI: `10.1109/LSP.2019.2943081`.

[7] Fridrik H. Z. Fridriksson. Bridging distributions for long term vessel prediction. An unpublished specialisation project, written in the spring of 2020 at the Norwegian University of Science and Technology, 2020.

[8] Simo Särkkä. *Bayesian Filtering and Smoothing*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013. DOI: `10.1017/CBO9781139344203`.

[9] Ronald P. S Mahler. *Statistical multisource-multitarget information fusion*. In Artech House, Boston, 2007. Chapter Appx D, pages 699–703. ISBN: 1-59693-093-4.

[10] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. 2011.

[11] The Society of Naval Architects and Marine Engineers. Nomenclature for treating the motion of a submerged body through a fluid. *Technical and Research Bulletin No.1-5*, 1950.

[12] Yaakov Bar-Shalom, Xiao-Rong Li, and Thiagalingam Kirubarajan. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Ltd, 2001, pages 269–270. ISBN: 9780471221272.

[13] B. I. Ahmad, J. K. Murphy, P. M. Langdon, S. J. Godsill, R. Hardy, and L. Skrypchuk. Intent inference for hand pointing gesture-based interactions in vehicles. *IEEE Transactions on Cybernetics*, 46(4):878–889, 2016.

[14] C. Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):234–236,395–404, 1978.

[15] Leonardo M Millefiori, Paolo Braca, Karna Bryan, and Peter Willett. Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction. eng. *IEEE transactions on aerospace and electronic systems*, 52(5):2313–2330, 2016. ISSN: 0018-9251.

[16] B. R. Dalsnes, S. Hexeberg, A. L. Flåten, B. H. Eriksen, and E. F. Brekke. The neighbor course distribution method with gaussian mixture models for ais-based vessel trajectory prediction:580–587, 2018.

[17] VT explorer. Ais ship types. 2020. URL: `https://api.vtexplorer.com/docs/ref-aistypes.html` (visited on 01/21/2021).

[18] International Maritime Organization. Ais transponders. 2019. URL: `https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx` (visited on 03/15/2021).

[19] The MathWorks. Latlon2local. 2021. URL: `https://se.mathworks.com/help/driving/ref/latlon2local.html` (visited on 03/15/2021).

[20] Erik F. Wilthil, Andreas L. Flåten, and Edmund F. Brekke. *A target tracking system for asv collision avoidance based on the pdaf*. In *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*. Thor I. Fossen, Kristin Y. Pettersen, and Henk Nijmeijer, editors. Springer International Publishing, Cham, 2017, pages 269–288. ISBN: 978-3-319-55372-6. DOI: `10.1007/978-3-319-55372-6_13`. URL: `https://doi.org/10.1007/978-3-319-55372-6_13`.

[21] The MathWorks. Chi2inv. 2021. URL: `https://se.mathworks.com/help/stats/chi2inv.html` (visited on 03/15/2021).

Appendices

A.I Examples of logarithmic representation

An example of logarithmic representation for some of the equations in section 3.2. As more measurements are sampled the likelihood values decreases, and to prevent underflow a natural logarithm is taken of the equations. For instance, taking the logarithm of the normal distribution is given by

$$\ln(\mathcal{N}(\mathbf{y}_k;\hat{\mathbf{y}}_k,\mathbf{J}_k)) = -\frac{1}{2}\left[(\mathbf{y}_k-\hat{\mathbf{y}}_k)^T\mathbf{J}^{-1}(\mathbf{y}_k-\hat{\mathbf{y}}_k)) + n\ln(2\pi) + \ln(|\mathbf{J}_k|)\right] \tag{8.1}$$

The logarithm of the numerical integral in eq. (3.8) is shown in eq. (8.2a). The terms $p(\mathbf{y}_{1:k}|t_{fi},d_n)$ $p(t_{fi}|d_n)$ are replaced with $g(t_{fi})$ for notational simplicity (eq. (8.3)). Also, the values inside the sum are written as the exponential of the logarithm, i.e. $e^{\ln(a)} = a$. To compute the exponential of the logarithms inside the brackets, without getting an underflow, a sufficiently large (or small) value c is chosen and each term is multiplied with $e^{c-c} = 1$, resulting in eq. (8.2c). The value of c could for instance be equal to the largest absolute value of $\ln(g(t_{fi}))$.

$$\begin{aligned}
\ln(p(\mathbf{y}_{1:k}|d_n)) &\approx \ln\Delta t_f \;+\; \ln\left[\frac{1}{2}e^{\ln(g(t_{f1}))} \;+\; \frac{1}{2}e^{\ln(g(t_{fq}))} \;+\; \sum_{i=2}^{q-1} e^{\ln(g(t_{fi}))}\right] && \text{(8.2a)}\\
&= \ln\Delta t_f \;+\; \ln\left[\frac{1}{2}e^{\ln(g(t_{f1}))}e^{c-c} \;+\; \frac{1}{2}e^{\ln(g(t_{fq}))}e^{c-c} \;+\; \sum_{i=2}^{q-1} e^{\ln(g(t_{fi}))}e^{c-c}\right] && \text{(8.2b)}\\
&= \ln\Delta t_f \;+\; c \;+\; \ln\left[\frac{1}{2}e^{\ln(g(t_{f1}))}e^{-c} \;+\; \frac{1}{2}e^{\ln(g(t_{fq}))}e^{-c} \;+\; \sum_{i=2}^{q-1} e^{\ln(g(t_{fi}))}e^{-c}\right] && \text{(8.2c)}
\end{aligned}$$

$$g(t_{fi}) := p(\mathbf{y}_{1:k}|t_{fi},d_n)p(t_{fi}|d_n) \tag{8.3}$$

Lastly, using this in eq. (3.10) where $p(\mathbf{y}_{1:k}|d)$ is replaced with $\exp(\ln(p(\mathbf{y}_{1:k}|d)))$, the weight on destination d_n can be written as eq. (8.4). Similarly as above, the equation is multiplied with e^{c-c} to prevent underflow.

$$\begin{aligned}
p(d_n|\mathbf{y}_{1:k}) &= \frac{\exp(\ln(p(\mathbf{y}_{1:k}|d_n)))p(d_n)}{\sum_{d\in\mathcal{D}}\exp(\ln(p(\mathbf{y}_{1:k}|d)))p(d)}\;\frac{\exp(-c)}{\exp(-c)}\\
&= \frac{\exp(\ln(p(\mathbf{y}_{1:k}|d_n)) - c)p(d_n)}{\sum_{d\in\mathcal{D}}\exp(\ln(p(\mathbf{y}_{1:k}|d)) - c)p(d)}
\end{aligned} \tag{8.4}$$

A.II Article's vs. Alternative mean: Destinations weights

The following graphs illustrate the difference of the destination weights estimation for BS-OU using the alternative mean and the equations given in [6].

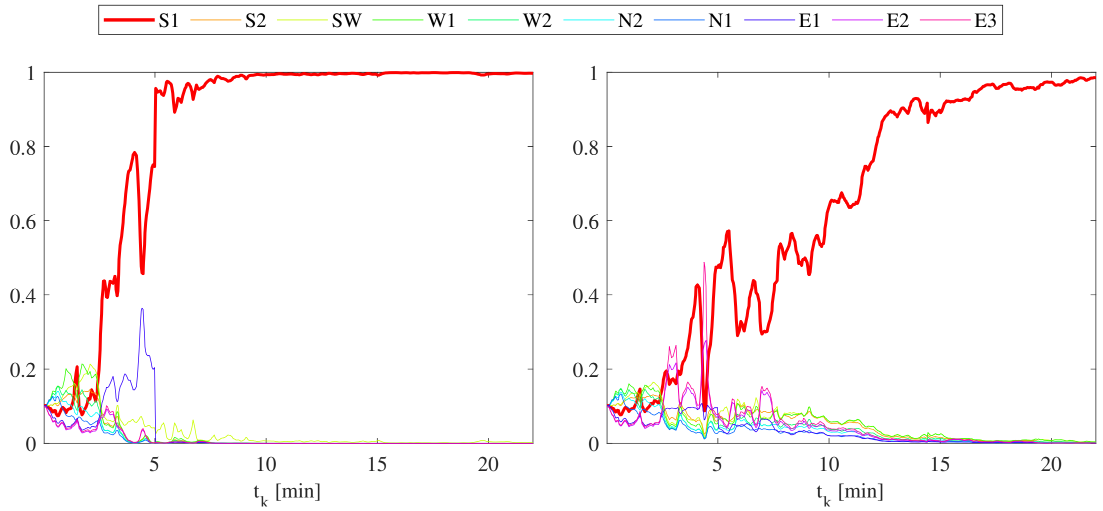


Figure A.1: Destination weights over time for *NE-S1* using BS-OU with, (*left:*) alternative mean and (*right:*) mean given in the article.

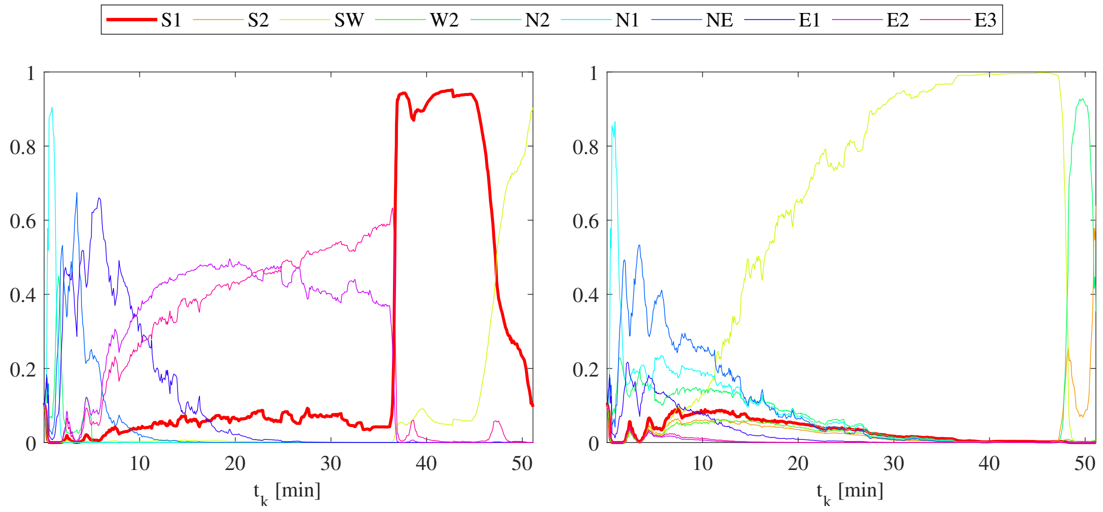


Figure A.2: Destination weights over time for *W1-S1* using BS-OU with, (*left:*) alternative mean and (*right:*) mean given in the article.

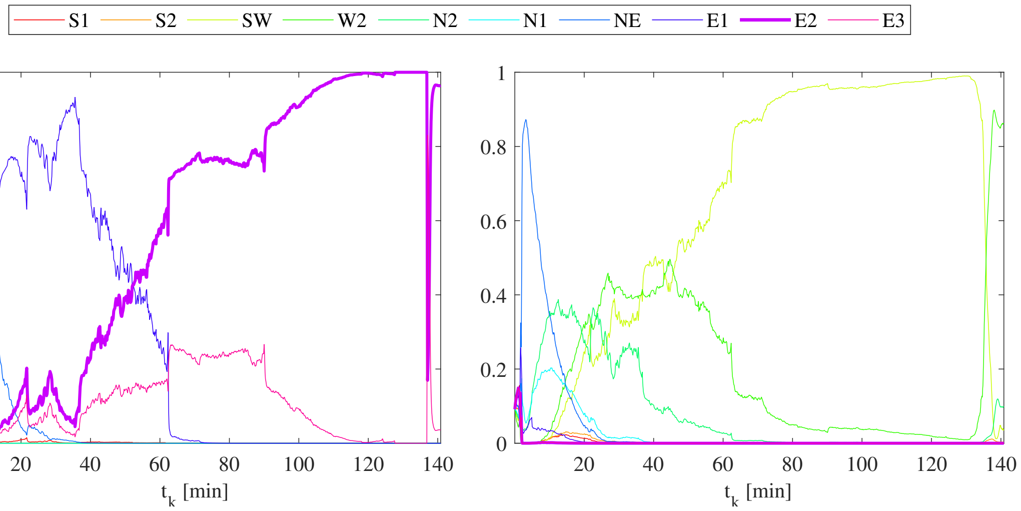


Figure A.3: Destination weights over time for *W1-E2* using BS-OU with, (*left:*) alternative mean and (*right:*) mean given in the article.

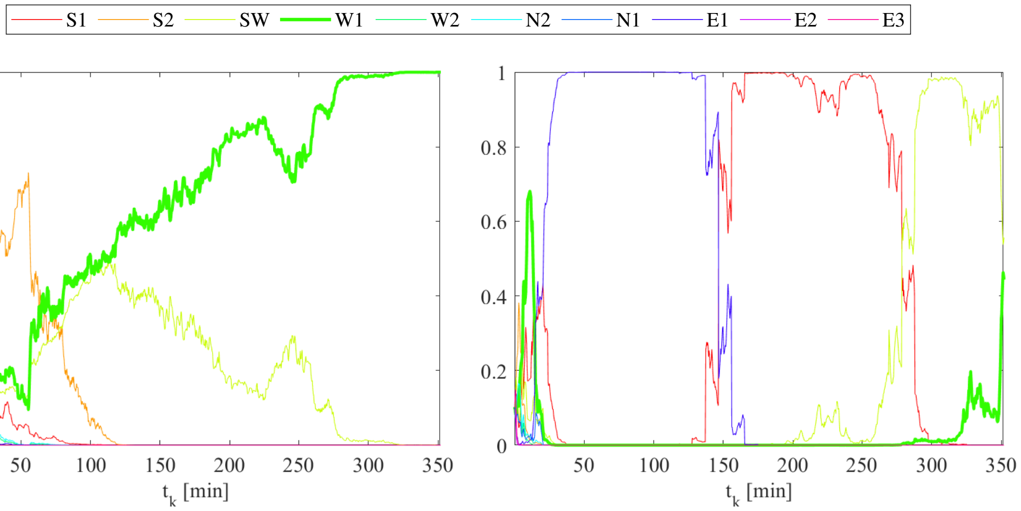


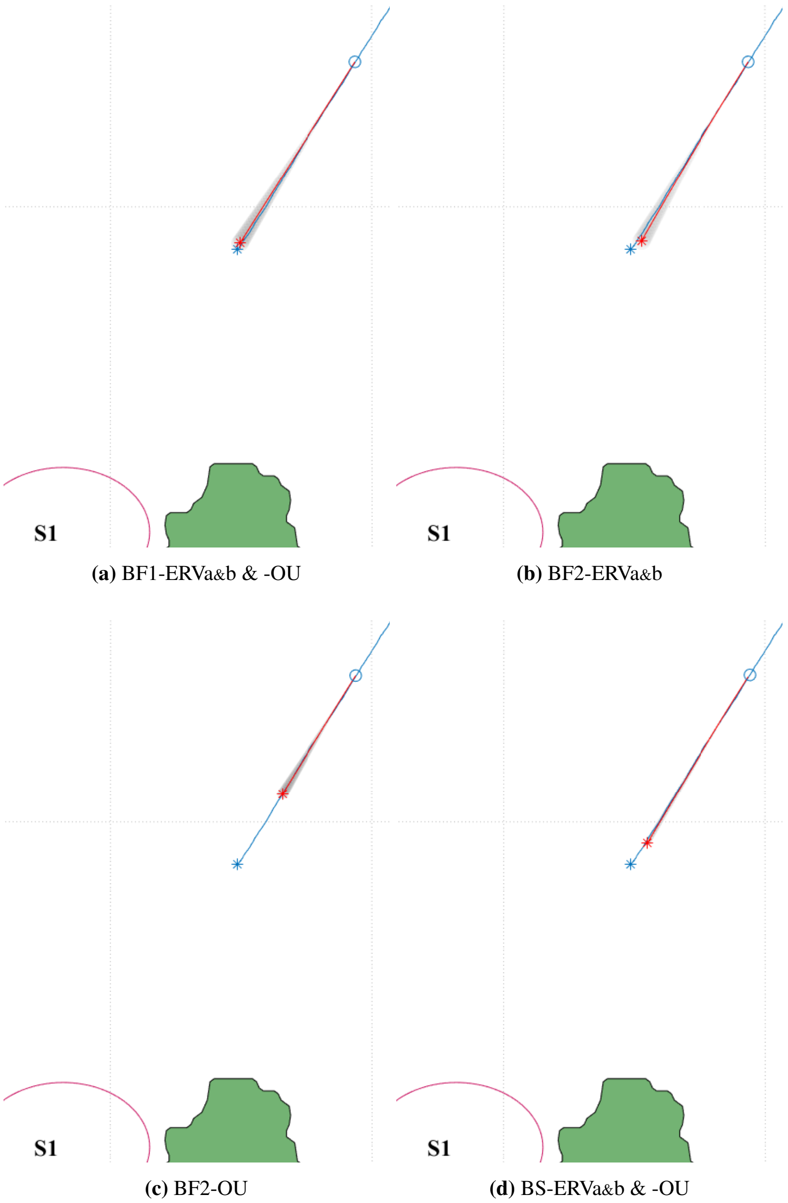
Figure A.4: Destination weights over time for *NE-W1* using BS-OU with, (*left:*) alternative mean and (*right:*) mean given in the article.

A.III Predicted trajectory

Additional illustrations of the predicted trajectories that were not shown in the main part of the thesis. Predictions are computed t_* minutes into the future from the current time t_k. The symbols used in the illustrations are given in the following table. Note that some illustrations have been rotated anticlockwise 90° to fit better on the page.

Symbol	Description
○	The current position at time t_k from which future trajectories are computed from
—	True measured trajectory
⋆	True position t_* minutes after the current time
⋆	Predicted position t_* minutes after the current time, for the destination MAP estimate
—	Predicted trajectory for the destination MAP estimate
●	Predicted position t_* minutes after the current time, for destination with weights over 0.1
●	Area of covariance

NE-S1



(a) BF1-ERVa&b & -OU

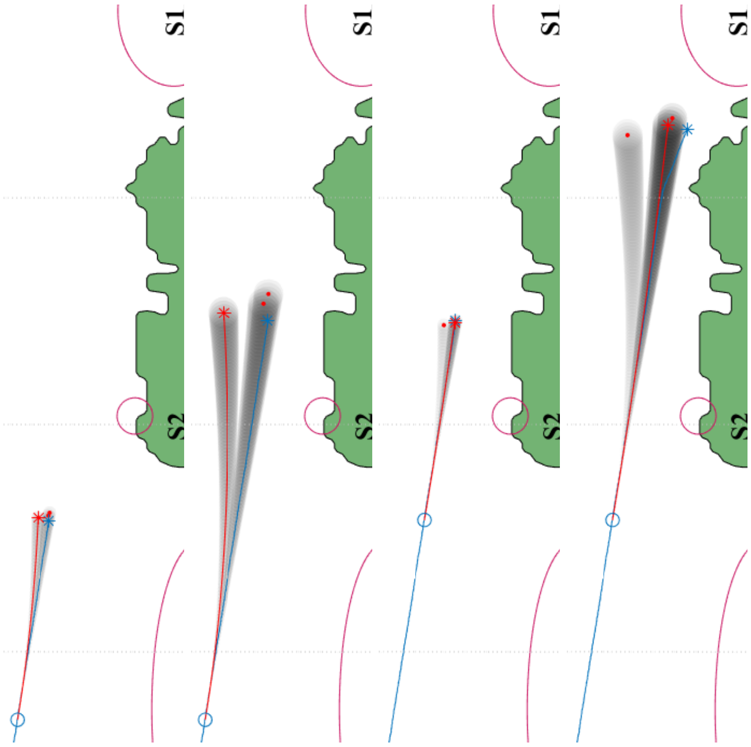
(b) BF2-ERVa&b

(c) BF2-OU

(d) BS-ERVa&b & -OU

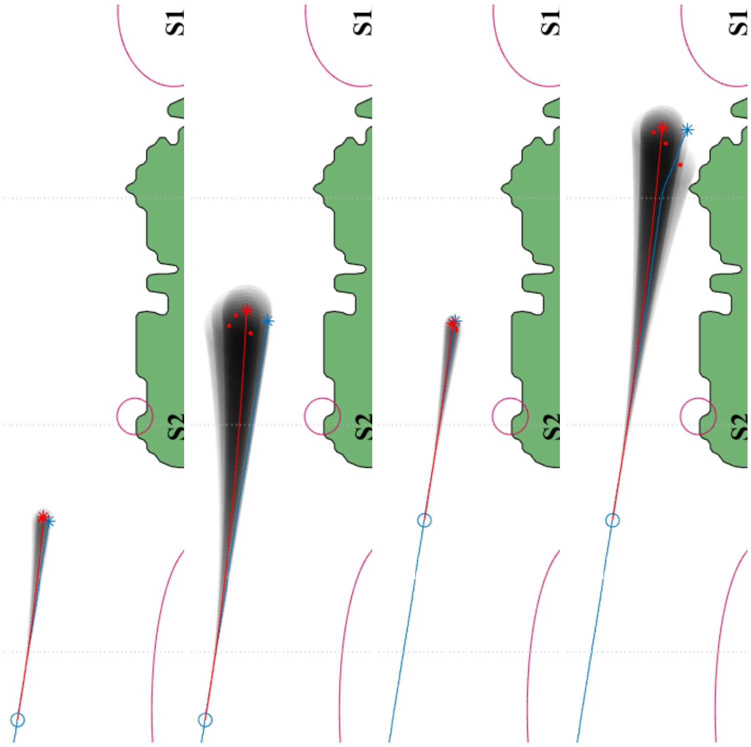
Figure A.5: *NE-S1*: $t_k = 10$ min, $t_* = 5$ min

W1-S1



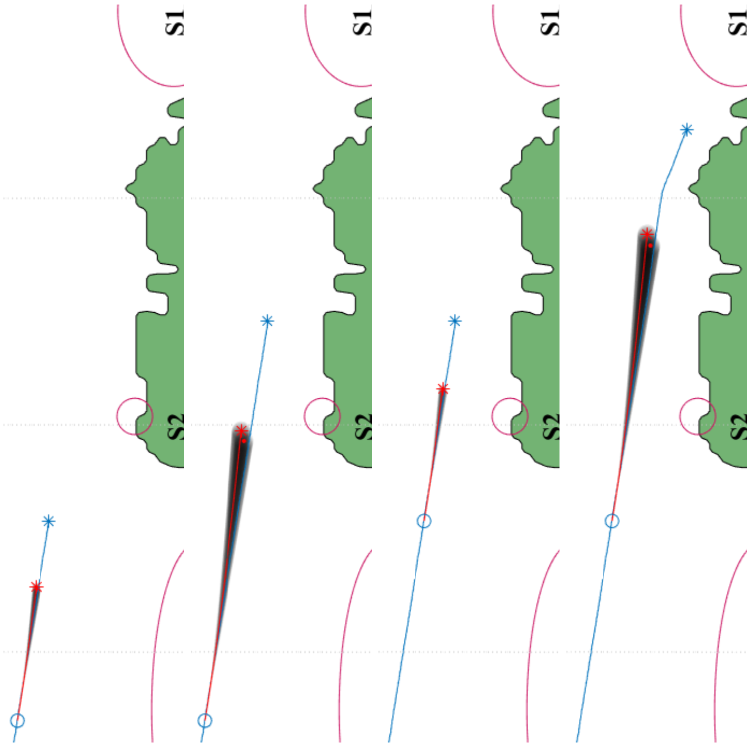
(a) $t_k = 10$ min, $t_* = 10$ min **(b)** $t_k = 10$ min, $t_* = 20$ min **(c)** $t_k = 20$ min, $t_* = 10$ min **(d)** $t_k = 20$ min, $t_* = 20$ min

Figure A.6: *W1-S1*: BF1-ERVa&b



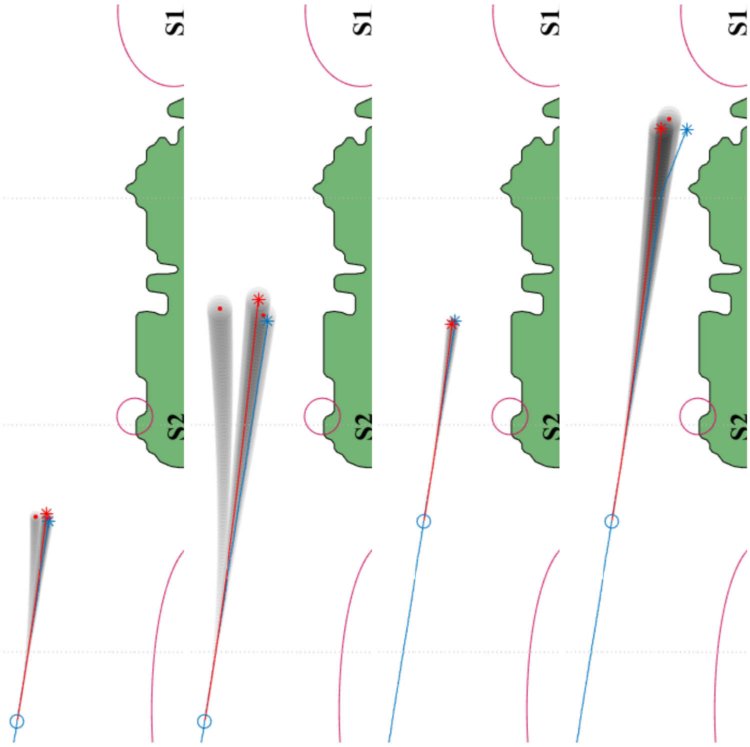
(a) $t_k = 10$ min, $t_* = 10$ min **(b)** $t_k = 10$ min, $t_* = 20$ min **(c)** $t_k = 20$ min, $t_* = 10$ min **(d)** $t_k = 20$ min, $t_* = 20$ min

Figure A.7: *W1-S1*: BF2-ERVa&b



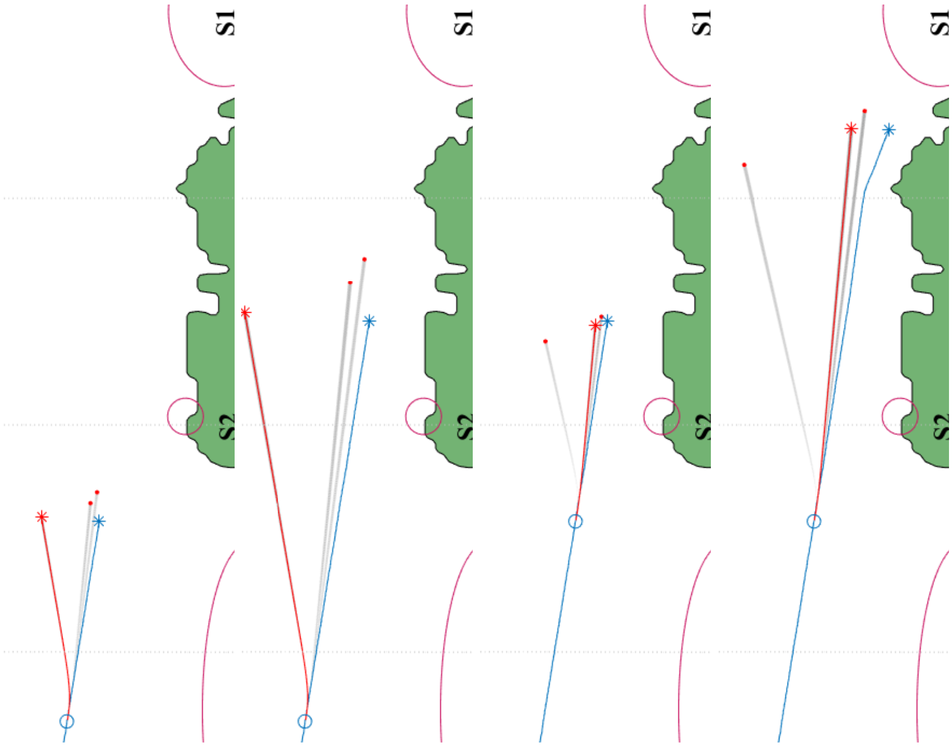
(a) $t_k = 10$ min, $t_* =$ 10 min **(b)** $t_k = 10$ min, $t_* =$ 20 min **(c)** $t_k = 20$ min, $t_* =$ 10 min **(d)** $t_k = 20$ min, $t_* =$ 20 min

Figure A.8: *W1-S1*: BF2-OU



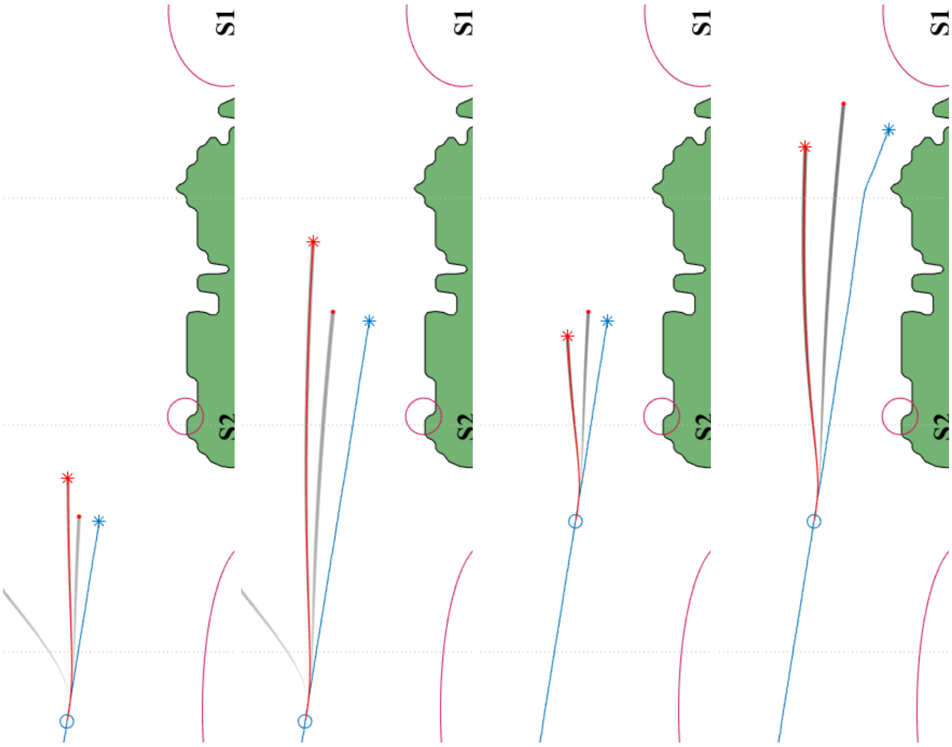
(a) $t_k = 10$ min, $t_* =$ 10 min **(b)** $t_k = 10$ min, $t_* =$ 20 min **(c)** $t_k = 20$ min, $t_* =$ 10 min **(d)** $t_k = 20$ min, $t_* =$ 20 min

Figure A.9: *W1-S1*: BF1-OU



(a) $t_k = 10$ min, $t_* = 10$ min **(b)** $t_k = 10$ min, $t_* = 20$ min **(c)** $t_k = 20$ min, $t_* = 10$ min **(d)** $t_k = 20$ min, $t_* = 20$ min

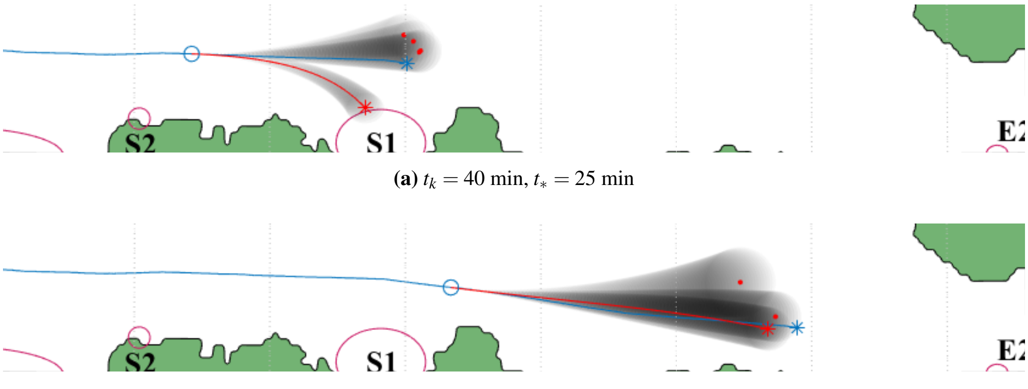
Figure A.10: *W1-S1*: BS-ERVa&b



(a) $t_k = 10$ min, $t_* = 10$ min **(b)** $t_k = 10$ min, $t_* = 20$ min **(c)** $t_k = 20$ min, $t_* = 10$ min **(d)** $t_k = 20$ min, $t_* = 20$ min

Figure A.11: *W1-S1*: BS-OU

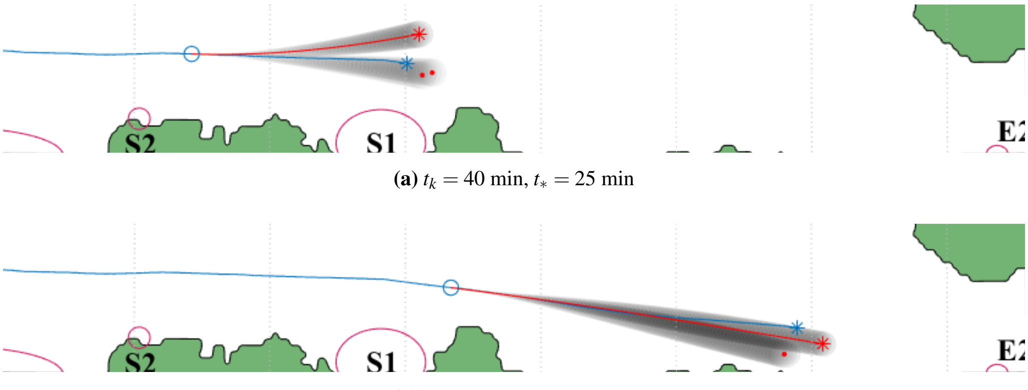
W1-E2



(a) $t_k = 40$ min, $t_* = 25$ min

(b) $t_k = 70$ min, $t_* = 40$ min

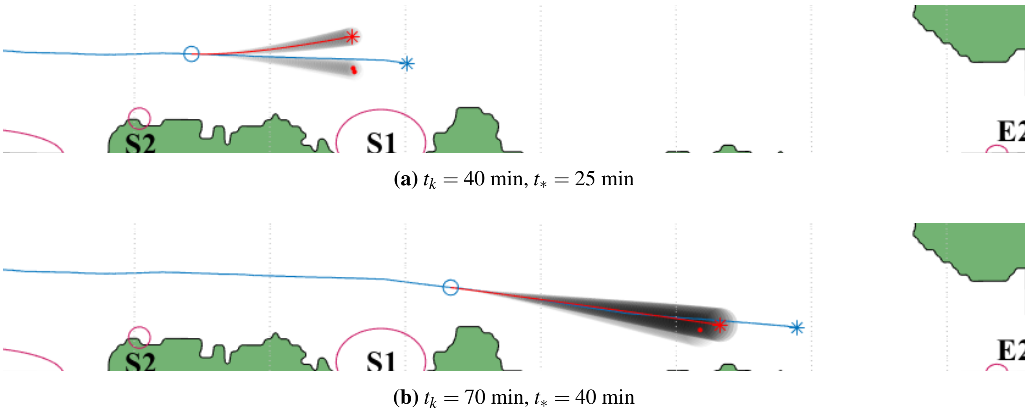
Figure A.12: *W1-E2*: BF2-ERVa



(a) $t_k = 40$ min, $t_* = 25$ min

(b) $t_k = 70$ min, $t_* = 40$ min

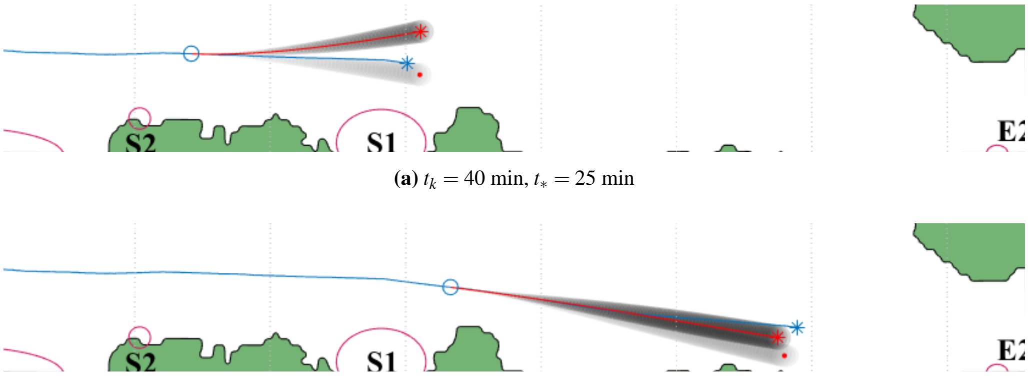
Figure A.13: *W1-E2*: BF1-ERVa



(a) $t_k = 40$ min, $t_* = 25$ min

(b) $t_k = 70$ min, $t_* = 40$ min

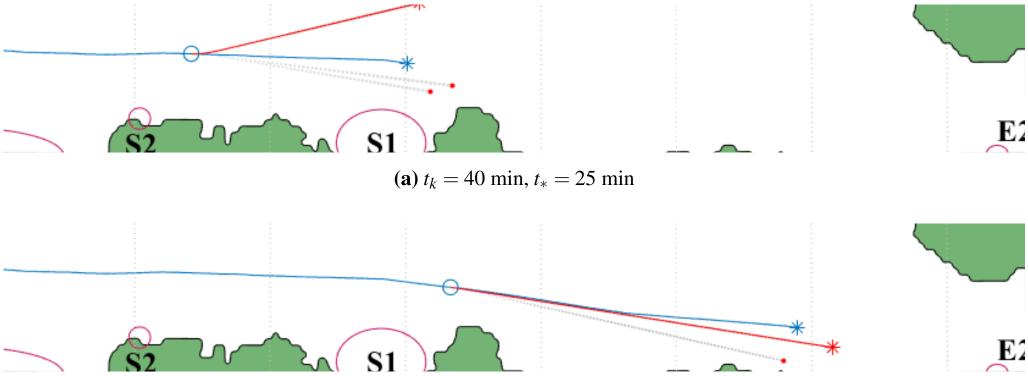
Figure A.14: *W1-E2*: BF2-OU



(a) $t_k = 40$ min, $t_* = 25$ min

(b) $t_k = 70$ min, $t_* = 40$ min

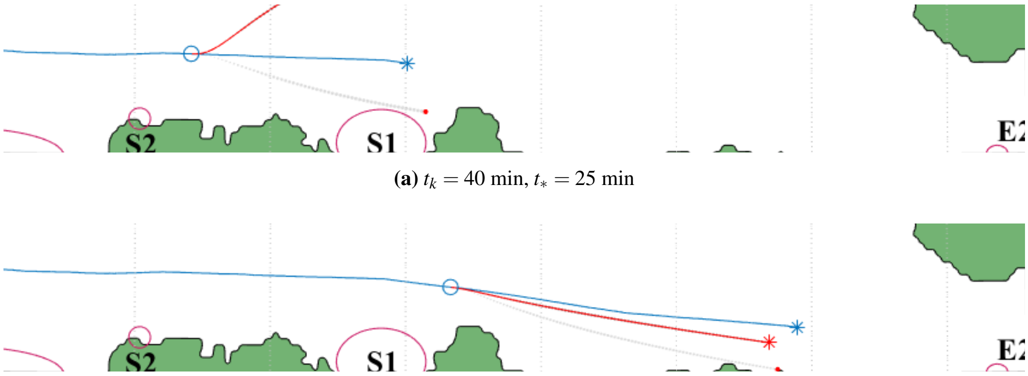
Figure A.15: *W1-E2*: BF1-OU



(a) $t_k = 40$ min, $t_* = 25$ min

(b) $t_k = 70$ min, $t_* = 40$ min

Figure A.16: *W1-E2*: BS-ERVa

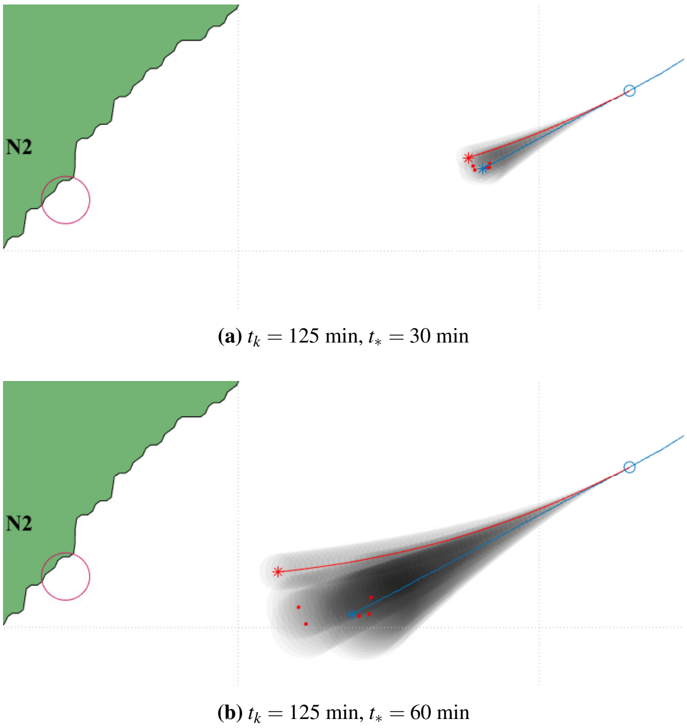


(a) $t_k = 40$ min, $t_* = 25$ min

(b) $t_k = 70$ min, $t_* = 40$ min

Figure A.17: *W1-E2*: BS-OU

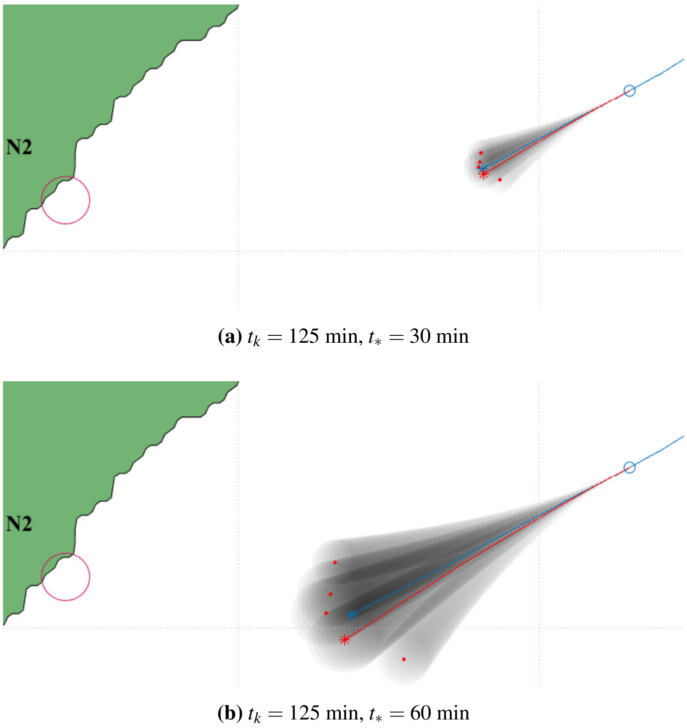
NE-W1



(a) $t_k = 125$ min, $t_* = 30$ min

(b) $t_k = 125$ min, $t_* = 60$ min

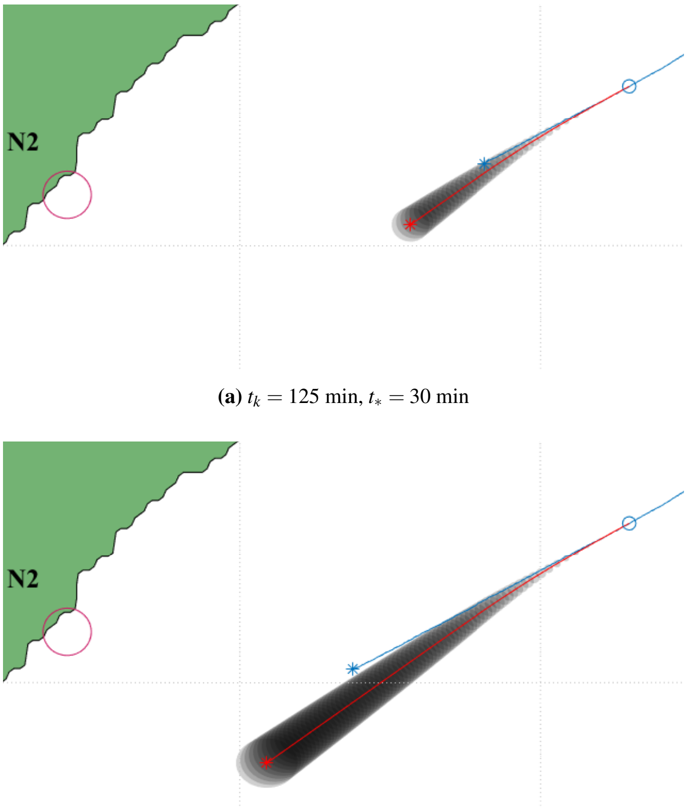
Figure A.18: *NE-W1*: BF2-ERVa



(a) $t_k = 125$ min, $t_* = 30$ min

(b) $t_k = 125$ min, $t_* = 60$ min

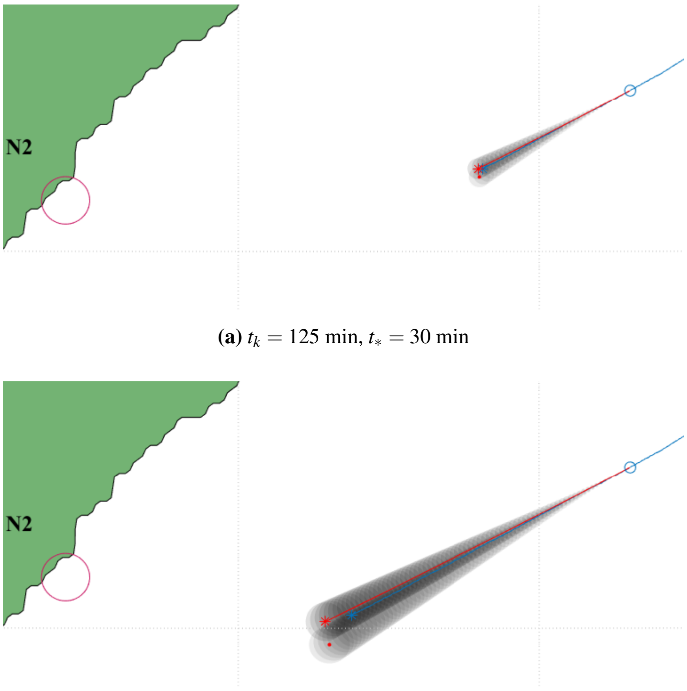
Figure A.19: *NE-W1*: BF1-ERVa



(a) $t_k = 125$ min, $t_* = 30$ min

(b) $t_k = 125$ min, $t_* = 60$ min

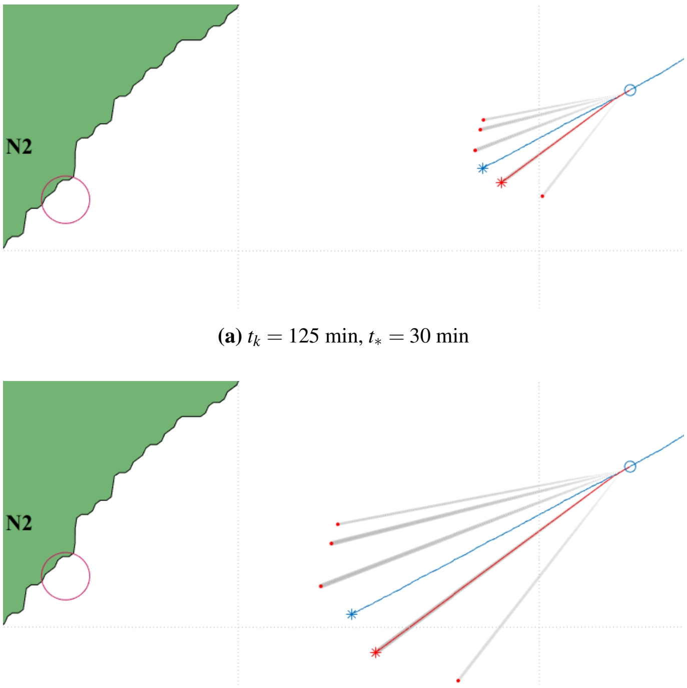
Figure A.20: *NE-W1*: BF2-OU



(a) $t_k = 125$ min, $t_* = 30$ min

(b) $t_k = 125$ min, $t_* = 60$ min

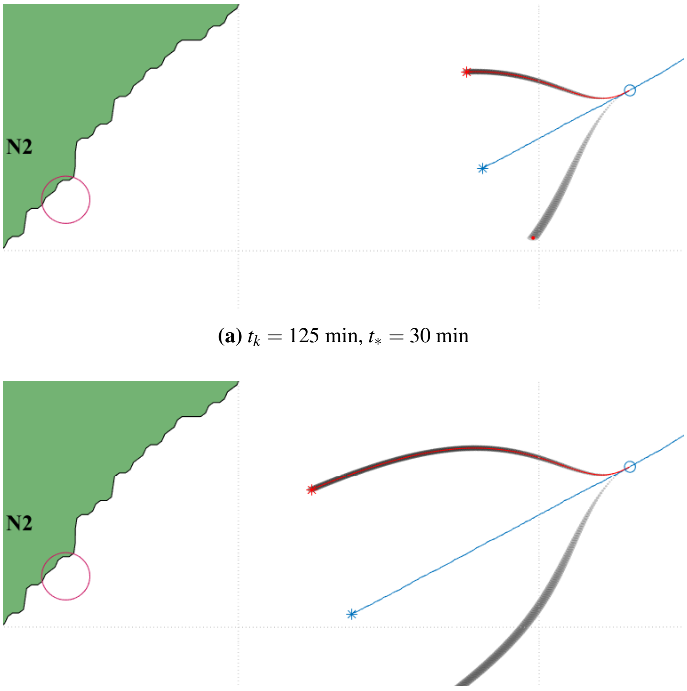
Figure A.21: *NE-W1*: BF1-OU



(a) $t_k = 125$ min, $t_* = 30$ min

(b) $t_k = 125$ min, $t_* = 60$ min

Figure A.22: *NE-W1*: BS-ERVa



(a) $t_k = 125$ min, $t_* = 30$ min

(b) $t_k = 125$ min, $t_* = 60$ min

Figure A.23: *NE-W1*: BS-OU

NTNU

Norwegian University of
Science and Technology