TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

SAKARI RAUTALIN

DATA-DRIVEN FORCE MODELS IN GNSS SATELLITE ORBIT PREDICTION

Master of Science thesis

Examiner: Phd. Simo Ali-Löytty and prof. Robert Piché
Examiner and topic approved by the Faculty Council of the Faculty of Natural Sciences
on 8th June 2016

ABSTRACT

SAKARI RAUTALIN: Data-driven force models in GNSS satellite orbit prediction
Tampere University of Technology
Master of Science thesis, 55 pages, 6 Appendix pages
March 2017
Master's Degree Programme in Science and Engineering
Major: Mathematics
Examiner: Phd. Simo Ali-Löytty and prof. Robert Piché
Keywords: GNSS, GPS, GLONASS, Beidou, latent force models, orbit prediction, SISRE, state estimation

In this study, we consider the problem of predicting the orbit of a GNSS satellite with a force model that can be adjusted based on data. In autonomous prediction, the goal is to use the positioning device in a completely or mostly autonomous mode. For this, the usable time of broadcast ephemeris need to be extended and to this end, predicting the orbit of a satellite is necessary. This is done by creating a force model for the satellite. In our previous research, the force model was based on four largest forces acting on a satellite: the gravitation of the Earth, the Sun and the Moon and solar radiation pressure. The position of the satellite in the future can then be computed by integrating the equation of motion with certain initial conditions.

The goal of this study was to improve the existing model by adding forces that can be estimated on the positioning device based on received data. This is done with latent force models, where additional forces have some prior model and need to be more accurately estimated with machine learning techniques. We create a state-space model for the latent forces, which is combined with the state-space model of the analytical motion of the satellite. Received broadcast ephemerides can then be used to estimate these forces in addition to position and velocity of a satellite. This is done as statistical inference with filtering and smoothing methods.

The main result of this study was that even a relatively simple model can improve prediction accuracy by a significant amount. After the largest forces have been taken into account, the largest improvement comes from a data-driven approach rather than adding more analytical terms to the force model. We created an adaptive algorithm, where data from a new broadcast can be used to update the estimates for the latent forces, which can then be used to predict the position of the satellite more accurately. Our model worked with all tested constellations: GPS, GLONASS and Beidou. The improvement was biggest with GPS and Beidou MEO satellites, while GLONASS satellites did not show as much improvement.

TIIVISTELMÄ

SAKARI RAUTALIN: Dataan perustuvat voimamallit GNSS satelliittien rataennustuksissa
Tampereen teknillinen yliopisto
Diplomityö, 55 sivua, 6 liitesivua
Maaliskuu 2017
Teknis-luonnontieteellinen koulutusohjelma
Pääaine: Matematiikka
Tarkastajat: TkT Simo Ali-Löytty ja prof. Robert Piché
Avainsanat: GNSS, GPS, GLONASS, Beidou, piilovoimamallit, rataennustus, SISRE, tilaestimointi

Tässä työssä tutkittiin paikannussatelliittien rataennustamista perustuen voimamalliin, jota voidaan muokata saadun datan perusteella. Satelliittien rataennustuksilla on useita käyttökohteita. Tässä tutkimuksessa rataennustusongelmaa tarkastellaan erityisesti autonomisen paikannuksen kannalta. Tämä on paikannustekniikka, jossa paikannuslaite pystyy itsenäisesti tai lähes itsenäisesti laskemaan satelliittien ratoja ja siten pidentämään satelliitista vastaanotettujen efemeridien käyttöikää. Tällöin ei ole tarvetta siirtää dataa esimerkiksi tukiasemien avulla.

Jo olemassa olevaa voimamallia pyrittiin parantamaan lisäämällä termejä, joiden luonteesta on olemassa jonkinlainen esitietämys, ja jota voidaan sitten tarkentaa saadun datan perusteella. Tämä tehdään piilovomamallien avulla. Piilovoimille luodaan tilamalli, joka yhdistetään osaksi liikeyhtälöiden avulla muodostettua satelliitin tilamallia. Näin saadaan tilamalli, joka nyt siis sisältää satelliitin paikan ja nopeuden lisäksi piilovoimakomponentit. Erilaisten suodatustekniikoiden sekä satelliitista saadun datan perusteella voidaan estimoida satelliitin paikka ja nopeus sekä piilovoimat suoraan paikannuslaitteessa.

Tutkimuksen tulos oli, että ennustustarkkuutta pystytään, tietyin oletuksin, parantamaan huomattavasti. Parannukset olivat jopa huomattavasti suurempia, joita saadaan pelkästään lisäämällä fysikaalisiin ilmiöihin perustuvia termejä voimamalliin. Työssä luotiin adaptiivinen algoritmi, joka päivittää piilovoimia satelliitista saadun datan perusteella. Näitä voidaan käyttää rataennustuksen aikana parantamaan ennustustarkkuutta. Sama malli toimi jokaisella testatulla satelliittikonstellaatiolla, joita olivat GPS, GLONASS sekä Beidou. Suurimmat parannukset saatiin GPS-satelliiteilla sekä Beidoun MEO-radoilla olevilla satelliiteilla, ja vastaavasti malli ei toiminut aivan yhtä hyvin GLONASS-satelliiteilla.

PREFACE

I started working in The Department of Mathematics in May 2014. From the beginning, I worked on topics for autonomous positioning, and have continued to do so for the past three years. My interest toward positioning applications and especially satellite-based positioning has increased considerably during this time.

The topic for this thesis arose in the spring of 2016, and I started the writing process during summer of 2016. I was, for a long time, rather sure that the topic for my Master's thesis would be related to satellite-based positioning and this topic proved to be very interesting and fruitful. Even though this is a thesis of mathematics, we do not go too deeply into mathematical theory behind the algorithms. This thesis should be read from the viewpoint of applied mathematics. The reader should be able to grasp the idea of this thesis even without significant mathematical background. A reader with more mathematical background may also want the view the more theoretical sections of this thesis.

I would like to thank my supervisors Simo Ali-Löytty and Robert Piché in the process of writing this thesis. I also want to express my gratitude to everyone in our positioning group who provided me with guidance or ideas not only in writing this thesis, but also during my time as part of the group and in autonomous prediction project.

For all the pleasures outside studies and work, I want to thank all my friends. My closest friends in the Guild of Science and Engineering Hiukkanen especially deserve my gratitude, for being an immense source of inspirational conversations and ideas and for providing numerous free-time activities during my studies at Tampere University of Technology.

I also want to thank my family: my mother, father, brother and sister for supporting me in every situation, and always encouraging to embrace my talent and interest towards mathematics and natural sciences. Last, I want to thank my girlfriend Anna for bringing stability into my life and for being there for me when I needed it the most.

Tampere, 21.3.2017

Sakari Rautalin

TABLE OF CONTENTS

LIST OF ABBREVIATIONS AND SYMBOLS

Abbreviations

BE	*Broadcast Ephemeris*
Beidou	*Beidou Weixing Daohang Xitong*
CTRS	*Conventional Terrestrial Reference System*
DE	*Development Ephemerides*
dUT1	Difference between UT1 and UTC
ECEF	*Earth-Centered Earth-Fixed*
EGM	*Earth Gravitational Model*
EKF	*Extended Kalman Filter*
EOP	*Earth Orientation parameters*
GEO	*Geostationary Orbit*
GLONASS	*Globalnaja Navigatsionnaja Sputnikovaja Sistema*
GNSS	*Global Navigation Satellite System*
GPS	*Global Positioning System*
HMM	*Hidden Markov model*
ICRS	*International Celestial Reference System*
IERS	*International Earth Rotation and Reference System Service*
IGS	*International GNSS Service*
IGSO	*Inclined Geosynchronous Orbit*
IRNSS	*Indian Regional Navigational Satellite System*
JPL	*Jet Propulsion Laboratory*
KF	*Kalman Filter*
LFM	*Latent Force Model*
MGEX	*Multi-GNSS Experiment*
MEO	*Medium-Earth Orbit*
PE	*Precise Ephemeris*
PRN	*Pseudo-Random Number*
QZSS	*Quasi-Zenith Satellite System*
RINEX	*Receiver Independent Exchange Format*
rms	*root mean square*
RTN	*Radial-Transverse-Normal*
SISRE	*Signal-In-Space-Range-Error*
SRP	*Solar Radiation Pressure*
TTFF	*Time-to-First-Fix*
UT1	*Universal Time*
UTC	*Coordinated Universal Time*

Notations

$t, u, \ldots$	scalars
$\mathbf{r}, \mathbf{v}, \ldots$	vectors
$\mathrm{F}, \mathrm{L}, \ldots$	matrices
$:=$	definition or notation
$\|\cdot\|$	vector (2-)norm
$p(x)$	probability distribution function of a random variable x.
$y\|x$	y with condition x
$\mathbf{x}_{1:k}$	values of $\mathbf{x}_i$ from $i = 1, \ldots, k$
$\mathbf{x}_0 \sim p(\mathbf{x}_0)$	variable $\mathbf{x}_0$ has probability distribution $p(\mathbf{x}_0)$
$\mathrm{N}(\mathbf{m}, \mathrm{P})$	multivariate normal distribution with mean $\mathbf{m}$ and covariance matrix P
$\mathbf{r}^T$	transpose of vector (or matrix)
A^{-1}	inverse of a matrix
$\mathrm{A}_{n\times m}$	matrix with n rows and m columns
c_n	variable c is function of integer variable n
$\mathbf{x}(t)$	variable $\mathbf{x}$ is function of real variable t
$\dot{\mathbf{r}}$	first derivative of $\mathbf{r}$ with respect to time
$\ddot{\mathbf{r}}$	second derivative of $\mathbf{r}$ with respect to time
$\sum_i$	summation of terms over i
∇	gradient
$x \in S$	x belongs to set S
Δx	error in variable x
$\mathrm{rms}(x)$	root mean square of variable x
$\overline{AB}$	vector from point A to point B

Symbols

$\mathbf{0}$	zero vector or matrix
A_k	transition matrix of linear dynamic model at time step k
A_{Sat}	area of a satellite
AU	average distance of the Earth from the Sun
a	ratio of the orbital period and one day
$\mathbf{a}$	acceleration of the satellite caused by analytical forces
$\mathbf{a}_a$	acceleration of the satellite caused by analytical and latent forces
$\mathbf{a}_{\mathrm{cb}}$	acceleration caused by the gravitation of a celestial body
$\mathbf{a}_i$	acceleration of the satellite caused by analytical force i
$\mathbf{a}_{\mathrm{Earth}}$	acceleration caused by the gravitation of the Earth
$\mathbf{a}_{\mathrm{latent}}$	acceleration caused by latent forces

$\mathbf{a}_{\text{Moon}}$	acceleration caused by the gravitation of the Moon
$\mathbf{a}_{\text{SRP}}$	acceleration caused by solar radiation pressure
$\mathbf{a}_{\text{SRP, direct}}$	solar radiation pressure acceleration to direction of the Sun
$\mathbf{a}_{\text{SRP, y-bias}}$	solar radiation pressure acceleration to y-bias direction
$\mathbf{a}_{\text{Sun}}$	acceleration caused by the gravitation of the Sun
α	elevation angle
α_1	parameter in SRP model
α_2	parameter in SRP model
G_k	variable in Extended fixed-point smoother -algorithm
b	constant bias term in the latent force model
$\mathbf{b}$	vector of RTN coordinate system constant accelerations
b_{N}	constant bias in latent forces to normal direction
b_{R}	constant bias in latent forces to radial direction
b_{T}	constant bias in latent forces to transverse direction
b_{X}	constant bias in latent forces to direction X
β	Brownian motion or azimuth angle
C_{nm}	coefficient of Earth Gravitational Model
c	speed of light
$\mathbf{c}$	vector of stochastic resonators and their derivatives
$\mathbf{c}_{\text{N}}$	normal direction stochastic resonator components and their derivatives
$\mathbf{c}_{\text{R}}$	radial direction stochastic resonator components and their derivatives
$\mathbf{c}_{\text{T}}$	transverse direction stochastic resonator components and their derivatives
$\mathbf{c}_{\text{X}}$	direction X stochastic resonator components and their derivatives
c_n	stochastic resonator
D	point of user location
D'	projection of user location to TN-plane
D	dispersion matrix
$\mathbf{e}$	error vector
$\mathbf{e}_{\text{Sat,Sun}}$	unit vector from the satellite to the Sun
$\mathbf{e}_{\text{y}}$	unit vector to y-direction
ε	leftover error of pseudorange measurement after clock and position errors
ϵ	noise term in the latent force model
ϵ_{N}	noise term of normal direction in the latent force model
ϵ_{R}	noise term of radial direction in the latent force model
ϵ_{T}	noise term of transverse direction in the latent force model
ϵ_{X}	noise term of direction X in the latent force model
$\mathbf{F}$	force exerted to the satellite
F	state transition matrix in latent force model

$\mathrm{F_N}$	state transition matrix of normal direction latent force component
F_n	state transition matrix of individual stochastic resonator
$\mathrm{F_R}$	state transition matrix of radial direction latent force component
$\mathrm{F_T}$	state transition matrix of transverse direction latent force component
$\mathrm{F_X}$	state transition matrix of direction X latent force component
$\mathrm{F}_{\mathbf{x}}$	Jacobian matrix of $\mathbf{f}$
$\mathbf{f}$	dynamic model function in a nonlinear state-space model
$\mathbf{f}_a$	augmented model function for the state of the satellite
f	base frequency of a stochastic resonator
G	gravitational constant
G_k	variable in Extended fixed-point smoother -algorithm
γ	radiation pressure coefficient
H	measurement matrix in the latent force model
H_k	measurement matrix of linear measurement model at time step k
$\mathrm{H_N}$	measurement matrix of normal direction in the latent force model
$\mathrm{H_R}$	measurement matrix of radial direction in the latent force model
$\mathrm{H_T}$	measurement matrix of transverse direction in the latent force model
$\mathrm{H_X}$	measurement matrix of direction X in the latent force model
$\mathrm{H}_{\mathbf{x}}$	Jacobian matrix of $\mathbf{h}$
$\mathbf{h}$	measurement model function in a nonlinear measurement model
I	identity matrix
K_k	variable in filtering algorithms
L	dispersion matrix in latent force model
L_a	dispersion matrix of the augmented model
L_n	dispersion matrix of individual stochastic resonator
$\mathbf{l}$	unit vector from satellite to the surface of the Earth
λ	longitude of the satellite
M_{cb}	mass of a celestial body
M_{E}	mass of the Earth
$\hat{\mathbf{m}}$	solution to differential equation in EKF
$\mathbf{m}$	mean of the distribution
$\mathbf{m}_k$	a posteriori estimate of the mean of the distribution describing the state
$\mathbf{m}_k^-$	a priori estimate of the mean of the distribution describing the state
m_{Sat}	mass of a satellite
$\boldsymbol{\mu}_k$	variable in EKF algorithm
N	normal direction, or position coordinate in this direction
N	number of components in the latent force model
N_{N}	number of components in the latent force model to normal direction
N_{R}	number of components in the latent force model to radial direction

N_{T}	number of components in the latent force model to transverse direction
n	integer variable
ν	shadow parameter
P	covariance matrix of the distribution
P_0	average solar radiation pressure at Earth's distance
P_k	a posteriori estimate of the covariance matrix of the distribution describing the state
P_k^-	a priori estimate of the covariance matrix of the distribution describing the state
P_{nm}	associated Legendre polynomials of degree n and order m
ϕ	latitude of the satellite
Q	covariance matrix
Q_β	diffusion matrix of Brownian motion β
$\mathrm{Q}_{\mathrm{latent}}$	covariance matrix of one latent force component
Q_k	covariance matrix of the process noise $\mathbf{q}_k$
q	variable in covariance matrix of latent forces
$\mathbf{q}$	Gaussian white noise
$\mathbf{q}_k$	process noise of dynamic model at time step k
q_n	spectral density of white noise $w_n(t)$
R_{E}	radius of the Earth
R	radial direction, or position coordinate in this direction
$\mathrm{R}_{\mathrm{ECEF}}$	transformation matrix from inertial coordinate system to ECEF
R_k	covariance matrix of the measurement noise $\mathbf{r}_k$
$\mathrm{R}_{\mathrm{RTN}}$	transformation matrix from inertial coordinate system to RTN coordinate system
r	distance of the satellite from the Earth's center
$\mathbf{r}_0$	initial position of a satellite
$\mathbf{r}_{\mathrm{cb}}$	position of a celestial body
$\mathbf{r}_k$	measurement noise of measurement model at time step k
$\mathbf{r}_{\mathrm{N}}$	unit vector in normal direction
$\mathbf{r}_{\mathrm{R}}$	unit vector in radial direction
$\mathbf{r}_{\mathrm{Sat}}$	position of a satellite
$\mathbf{r}_{\mathrm{Sun}}$	position of the Sun
$\mathbf{r}_{\mathrm{T}}$	unit vector in transverse direction
$\mathbf{r}$	position of the satellite in inertial coordinate system
ρ	pseudorange measurement
S	point of satellite's true position
S'	point of satellite's predicted position
S_k	variable in filtering algorithms
S_{nm}	coefficient of Earth Gravitational Model
σ	standard deviation

σ^2	variance
T	transverse direction, or position coordinate in this direction
T_{orb}	orbital period of a satellite
t	time (as a function variable)
t_0	specific time instant (initial time of integration etc.)
t_{toe}	time of ephemeris
t_u	user clock offset
δt	satellite clock offset
θ	angle between the center of the Earth and the edge of the Earth viewed from the satellite
U_{E}	potential of the Earth's gravitational field
u	modelled latent force
u_{N}	normal direction component of the latent forces
u_{R}	radial direction component of the latent forces
u_{T}	transverse direction component of the latent forces
u_{X}	latent force component to direction X
$\mathbf{u}$	position of the user
$\mathbf{v}_0$	initial velocity of a satellite
$\mathbf{v}_k$	variable in Kalman filter algorithm
$\mathbf{v}$	velocity of the satellite in inertial coordinate system
$\mathbf{w}$	vector of stochastic resonator noise terms
$\mathbf{w}_a$	augmented noise term for latent force model
$\mathbf{w}_b$	noise of the constant acceleration terms
$\mathbf{w}_{\text{N}}$	noise of latent force in normal direction
w_n	noise term of stochastic resonator
$\mathbf{w}_{\text{R}}$	noise of latent force in radial direction
w_{R}	coefficient in the calculation of SISRE
$\mathbf{w}_{\text{T}}$	noise of latent force in tangental direction
$w_{\text{T,N}}$	coefficient in the calculation of SISRE
$\mathbf{w}_{\text{X}}$	noise of latent force in direction X
$\mathbf{w}_{\mathbf{x}}$	noise term in the stochastic differential equation of the state of the satellite
X	subscript for direction
$\mathbf{x}$	state in state-space model or state of the satellite consisting of position and velocity
$\mathbf{x}_0$	initial state
$\mathbf{x}_a$	augmented state of the satellite consisting of position, velocity and latent forces
$\mathbf{x}_k$	state at time step k
$\mathbf{y}_k$	measurement at time step k
Φ	matrix in EKF algorithm
$\hat{\Phi}$	solution to differential equation in EKF algorithm

1. INTRODUCTION

One of the most important methods, when considering navigation and positioning applications, is satellite-based positioning. *Global Navigation Satellite System* (GNSS) is a constellation of satellites, which is especially designed for positioning purposes. Most people are familiar with GPS (*Global Positioning System*), but other systems are also available.

Positioning is based on pseudorange measurements. These measurements are of form

$$\rho = \|\mathbf{u} - \mathbf{r}\| + c(t_u - \delta t) + \varepsilon, \tag{1.1}$$

where $\mathbf{u}$ is the user's position, $\mathbf{r}$ is the satellite's position, t_u is the user clock offset, δt is the satellite clock offset, and c is the speed of light. Term ε is an additional error term, which includes, among others, atmospheric delays and multipath errors. Term pseudorange is used because we do not only measure the difference in position but we also include the effect of time in the equation.

Satellite positions and clock offsets can be calculated from the data sent by the satellite. User position $\mathbf{u}$ is a 3-dimensional vector, which we wish to calculate. In addition to this, the user clock offset needs to be calculated. This means that we have 4 variables to be calculated and therefore we need data from at least 4 satellites. The received data can then be used to calculate the user's position from a set of equations of form (1.1).

The data from the satellite, which in addition to position and clock offset includes for example satellite identification data and the health of the satellite, is transmitted in a data structure called *Broadcast Ephemeris* (BE). This set of data has some validity time, outside of which the data can no longer be used to calculate the user's position, and we need to receive a new ephemeris set from the satellite so it can be used for positioning. Receiving the data takes some time, for example it takes 18 seconds for GPS satellite to send the necessary information for positioning purposes. Additionally, sending this data is repeated only once every 30 seconds. If the user is for example in an urban environment, where buildings can interfere with the signal, it may take even several minutes to receive the entire signal. From the point of view of the user, this is an an annoying feature and in some applications it can be costly

or even fatal. [32]

Autonomous positioning aims to extend the usable time of the ephemeris. This allows BE data to be used beyond its validity time, and therefore *Time to First Fix* (TTFF), which is the time from when device is turned on until the positioning device can give any estimate of the position, can be reduced to about 5 seconds [32]. Autonomous positioning can also be used in other ways to widen the capability of GNSS devices. For example, something can block the signal partially, such that the signal is strong enough for the pseudorange to be detected, but not strong enough for reading the entire ephemeris. In this case, older ephemerides can be used once their validity time has been extended.

One goal in autonomous positioning is that it can work on a device that is completely lacking the possibility of a network connection. This is, however, rarely the case and usually we can deliver some data to the device as assistance data, which can be calculated on a server that can use any data available. The data to be delivered can be for example some satellite-specific parameters, such as antenna correction parameters, which can be used to improve position estimates from the satellite. We assume in this work that the device will work in a semi-autonomous way; it needs to predict the satellite ephemerides, but small amount of assistance data can also be delivered as occasionally.

Assume now that we have received a broadcast ephemeris from the satellite and we would like to extend its validity period. It is possible, based on the received data, to predict the position and clock offset values in the future. This prediction can then be used as prior information to reduce TTFF. In this work, we will focus on predicting the position of the satellite. This is done by estimating the satellite's initial position and velocity from available data, and then integrating the equation of motion of the satellite to estimate these at a later time. Our research group has published several papers on methods for reducing the prediction error [9, 28, 32].

When we want to compare the accuracy of our prediction to the true position, we would like the reference data to be as accurate as possible. To achieve this goal, we shall use position data from *Precise Ephemeris* (PE). This is centimeter-level accurate positions of satellites calculated with high computation power and laser ranging measurements. This is not usually available to positioning devices and hence it can't be used in this sort of application, but it gives a good reference for accuracy comparison when considering these kinds of methods. Additionally, we want to view the prediction error from the point of view of positioning applications, so we should also clarify, how the error in the satellite's position affects the pseudorange measurement (1.1).

Latent force models have been studied by Alvarez et. al. in [10, 11, 12] and by Särkkä

et. al. in [17, 18]. In their studies, physical systems are interpreted with latent forces, which essentially are forces that are based on some prior knowledge, and then are estimated more precisely based on data. The most important publication related to our studies is [18], where latent forces are used to improve satellite orbit predictions.

The main purpose of this study is to find out, if data received from the satellite can be used to estimate latent forces, which could possibly be used to make our predictions more accurate. We have found that our prediction errors show systematic behaviour, which suggests that our model has incorrect or missing forces. We can therefore assume that the prediction error can be reduced by adding additional terms to the force model. The additional forces are added as latent forces rather than physics-based expressions of known forces. In short, we attempt to fix our force model with machine learning techniques based on the data about true orbit, which is received via BE from the satellite.

In Chapter 2, we give an introduction to filtering theory, which will be necessary when we want to estimate the missing forces. In Chapter 3, we present the model for our orbit, which includes the force model, necessary coordinate systems and our initial state estimation algorithm. Chapter 4 focuses on latent force models and how these can be incorporated in satellite orbit prediction. In Chapter 5, we present various GNSS, and describe how we will measure the accuracy of the predictions. We present the results of this study in Chapter 6 and give conclusions in Chapter 7.

2. FILTERING THEORY

We shall start by introducing a few concepts of probability theory, state estimation theory and measurement theory. The theorems and definitions are presented directly with vectors and we note that corresponding theorems with scalar variables are only special cases of the theorems below. Furthermore, when we discuss probability distribution functions in this study, we always talk about realizations of random variables. For example, with $p(\mathbf{x})$ we mean the probability that the realization of the corresponding random variable is $\mathbf{x}$. For probability theory, or for state-space and measurement theory, one may refer to [19] or [29].

Theorem 2.1 (Bayes' law)

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})},$$

where $\mathbf{x}$ *and* $\mathbf{y}$ *are realizations of the corresponding random variables.*

Here, probability distribution p is a multivariate probability distribution, since inputs are vectors.

The inputs $\mathbf{x}$ and $\mathbf{y}$ have been chosen for a specific reason. In this work, and more generally in measurement and state-space theory, $\mathbf{x}$ is used to notate the state while $\mathbf{y}$ is used to notate measurements. Bayes' law in the form above then gives us means to estimate the probability distribution of the state based on measurements.

In this work, we use a *Hidden Markov model* (HMM). This means that in our sequence of states $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$, the states are hidden and can't be directly observed. We also have a sequence of observations $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$, which are called measurements, and are linked to states according to some model. Both of these are time-series, and subscript indicates the timestep.

When we use HMM, the state and measurement models have useful properties, which become important when constructing state estimation algorithms. These are *Markov property of states* and *Conditional independence of measurements.*

Property 2.1 (Markov property of states) *States* $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ *form a Markov sequence. Markov property of this sequence means that state* $\mathbf{x}_k$ *depends only on the preceding state* $\mathbf{x}_{k-1}$ *and is independent of any state that has occurred before this. This means*

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

Property 2.2 (Conditional independence of measurements) *In a sequence of measurements* $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$ *of the states* $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$*, the measurements are conditionally independent. This means that*

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

Theorem 2.1 and Properties 2.1 and 2.2 allow us to form a recursive solution the state estimation problem.

2.1 Filtering problem

In this work, we shall consider probabilistic state-space models. This means that the initial state, dynamic model and measurement model are considered as probability distributions. Hence, the problem can be written in general form as

$$\mathbf{x}_0 \sim p(\mathbf{x}_0), \tag{2.1a}$$
$$\mathbf{x}_k \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}), \tag{2.1b}$$
$$\mathbf{y}_k \sim p(\mathbf{y}_k|\mathbf{x}_k). \tag{2.1c}$$

The notation above means that states and measurements are modelled as probability distributions. We note that equations (2.1) do not describe the full distribution based on all previous states and measurements. It means that our state-space model is based on the properties of HMM: our initial state has some probability distribution, each state is only dependent on the preceding state (Property 2.1) and each measurement is only dependent on the corresponding state (Property 2.2). From this it follows that we can sequentially calculate our states $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ once we have received measurements $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$. These are reasonable assumptions for our application of estimating the orbit of a satellite and they allow us to use simpler methods for solving the state estimation problem.

In many applications only a point-estimate of the state is required. For example, in our context, when we predict the orbit of a satellite for positioning purposes, we are basically only interested in the best estimate of the position and velocity and not

so much in the probability distribution. When we are, however, constructing state estimation algorithms, we are interested in estimating not only the state but also the uncertainty of the estimate. Therefore, we will model the states with probability distributions.

We use another assumption: the errors in dynamic model and measurement model have Gaussian distributions. In many cases this is a sufficiently good approximation. Also, when we use this approximation, in the linear case, the mean of the distribution is the optimal estimate in the sense that it minimizes the mean square error of the estimate.

Filtering algorithms usually consist of two steps: prediction step and update step. In prediction step we want to estimate the next state of the satellite based on previous ones. Hence we get *a priori* estimate for the probability distribution. In update step, in addition to the predicted state, we use the received measurement to estimate the state more accurately. Here we calculate *a posteriori* estimate for the probability distribution. If we lack measurement of the corresponding timestep, this step is omitted and predicted states are used as *a posteriori* estimates.

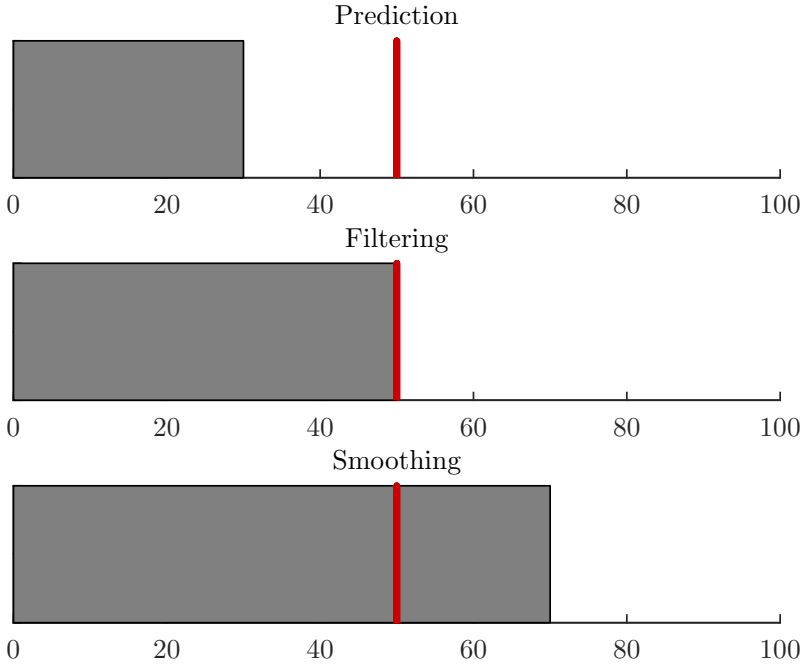

Figure 2.1 *Figure illustrating the difference between prediction, filtering and smoothing. In this example, we are interested in state at timestep $k = 50$ (red line), and we have measurements $y_{1:n}$ (grey area).*

We further clarify the terminology used here to distiguish three different cases. We wish to estimate the state $\mathbf{x}_k$ based on measurements $(\mathbf{y}_1, \ldots, \mathbf{y}_n)$. We shall use the

following terminogy:

1. prediction, when $k > n$,
2. filtering, when $k = n$ and
3. smoothing, when $0 \leq k < n$.

In Figure 2.1 we illustrate the difference between the three.

Before we consider any filtering algorithms, we shall introduce a few different state-space models.

2.2 State-space models

Here we shall take a look at some state-space and measurement models. As stated earlier, error distributions are considered to be Gaussian.

State-space models usually consists of two different equations: One is the dynamic model and the other is the measurement model. Dynamic model tells about the evolution of the model, for example the relation between two consecutive states. This is where Property 2.1 is needed to create simpler model. Measurement tells the relation between the state and measurement and this is where Property 2.2 allows us to further simplify the model.

We shall consider both continuous and discrete state-space models. In this context, continuous and discrete refer to time domain and not values of the random variables in the model. The models are named after the continuity of their dynamic and measurement models. For example, models where the measurement model and the dynamic model are both discrete, are called discrete-discrete state-space models.

We shall start with the simplest example: linear state-space model.

Definition 2.1 (Linear state-space models) *Dynamic model and measurement model are linear Gaussian if*

$$\begin{aligned} \mathbf{x}_k &= \mathrm{A}_{k-1}\mathbf{x}_{k-1} + \mathbf{q}_{k-1} \\ \mathbf{y}_k &= \mathrm{H}_k\mathbf{x}_k + \mathbf{r}_k, \end{aligned}$$

where A_{k-1} *is the transition matrix of the dynamic model,* H_k *is the measurement matrix of the measurement model,* $\mathbf{q}_{k-1} \sim \mathrm{N}(\mathbf{0}, \mathrm{Q}_{k-1})$ *is process noise and* $\mathbf{r}_k \sim \mathrm{N}(\mathbf{0}, \mathrm{R}_k)$ *is measurement noise.*

In linear case, we use only linear transformation, i.e. matrix multiplication and summation of Gaussian distributions. Therefore, the resulting distribution is also Gaussian.

Next we shall consider nonlinear models.

Definition 2.2 (Nonlinear state-space model) *Dynamic model and measurement model are nonlinear if*

$$\begin{aligned}\mathbf{x}_k &= \mathbf{f}_{k-1}(\mathbf{x}_{k-1}) + \mathbf{q}_{k-1}\\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k) + \mathbf{r}_k,\end{aligned}$$

where $\mathbf{f}_{k-1}$ *is the dynamic model function,* $\mathbf{h}_k$ *is the measurement function and* $\mathbf{q}_{k-1}$ *and* $\mathbf{r}_k$ *are the same noise terms as in linear state-space models.*

In the nonlinear case, the resulting distribution is not necessarily Gaussian. This is, however, sometimes assumed to design simpler filtering algorithms. This is for example the case in Extended Kalman filter, where the models are linearized and therefore equations result in Gaussian distributions.

We shall consider one more model, which is the same type as in Definition 2.2, except that dynamic model is continuous in time.

Definition 2.3 (Continuous-discrete state-space models) *A nonlinear state-space model is continuous-discrete if dynamic model and measurement model are*

$$\begin{aligned}\frac{d\mathbf{x}(t)}{dt} &= \mathbf{f}(\mathbf{x}(t), t) + \mathbf{q}(t)\\ \mathbf{y}_k &= \mathbf{h}_k(\mathbf{x}_k) + \mathbf{r}_k,\end{aligned}$$

where $\mathbf{h}_k$ *and* $\mathbf{r}_k$ *are the same as in Definition 2.2. The state* $\mathbf{x}(t)$*, dynamic model function* $\mathbf{f}(\mathbf{x}(t), t)$ *and noise* $\mathbf{q}(t)$ *are now in continuous time domain instead of discrete. The noise* $\mathbf{q}(t)$ *is Gaussian white noise with spectral density matrix* Q*.*

It should be noted that discrete versions of state-space models may also contain time-dependence. This is notated with subscript k, which refers to the current timestep t_k.

State-transition model of Definition 2.3 can also be presented as Itô stochastic differential equation. If we formally multiply with dt and additionally use linear transformation for the noise, we get

$$d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t)dt + \mathrm{D}(\mathbf{x}(t), t)d\beta, \tag{2.2}$$

where now β describes Brownian motion with diffusion matrix Q_β and D is the dispersion matrix. This is the corresponding Itô stochastic differential equation.

Continuous models are in this work handled by discretization. This means that model is converted into discrete-version by using some timestep and then appropriate functions are chosen to correspond to the timestep of the discretized model.

2.3 Solutions to filtering and smoothing problems

The problem we are especially interested in is estimating the state of the satellite based on measurements from the satellite broadcast. We shall proceed towards this goal progressively by introducing different filters, which in the end are suitable for this purpose. First we shall look at the most simple example.

2.3.1 Kalman filter

When the dynamic model and measurement model are linear and Gaussian as in Definition 2.1, it is possible to derive a closed-form optimal solution to filtering problem. This algorithm is known as *Kalman filter* (KF).

As we stated earlier, linearity of the dynamic and measurement model lead to the resulting state probability distributions also being Gaussian. This is considered in Kalman filter, where the mean and covariance matrix of the state are estimated.

Algorithm 2.1 *If the dynamic model and the measurement model are according to Definition 2.1, then predicted mean and covariance (a priori) are*

$$\begin{aligned}\mathbf{m}_k^- &= \mathrm{A}_{k-1}\mathbf{x}_{k-1},\\ \mathrm{P}_k^- &= \mathrm{A}_{k-1}\mathrm{P}_{k-1}\mathrm{A}_{k-1}^T + \mathrm{Q}_{k-1},\end{aligned}$$

and using measurement $\mathbf{y}_k$*, mean and covariance (a posteriori) are updated with*

$$\begin{aligned}\mathbf{v}_k &= \mathbf{y}_k - \mathrm{H}_k\mathbf{m}_k^-,\\ \mathrm{S}_k &= \mathrm{H}_k\mathrm{P}_k^-\mathrm{H}_k^T + \mathrm{R}_k,\\ \mathrm{K}_k &= \mathrm{P}_k^-\mathrm{H}_k^T\mathrm{S}_k^{-1},\\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathrm{K}_k\mathbf{v}_k,\\ \mathrm{P}_k &= \mathrm{P}_k^- - \mathrm{K}_k\mathrm{S}_k\mathrm{K}_k^T.\end{aligned}$$

Proof. Omitted. Can be found for example in [29, p. 57-58].

The form given for Kalman filter here is not the only possible one. According to Särkkä in [29, p. 58], it would be better to work with matrix square roots of covariances instead of covariances plainly to avoid numerical errors caused by the matrix inversion. This should especially be considered in computer simulations and numerical implementations. We also use this approach in this work, although in the context of Extended Kalman filter and use Cholesky factorization to form the matrix square roots.

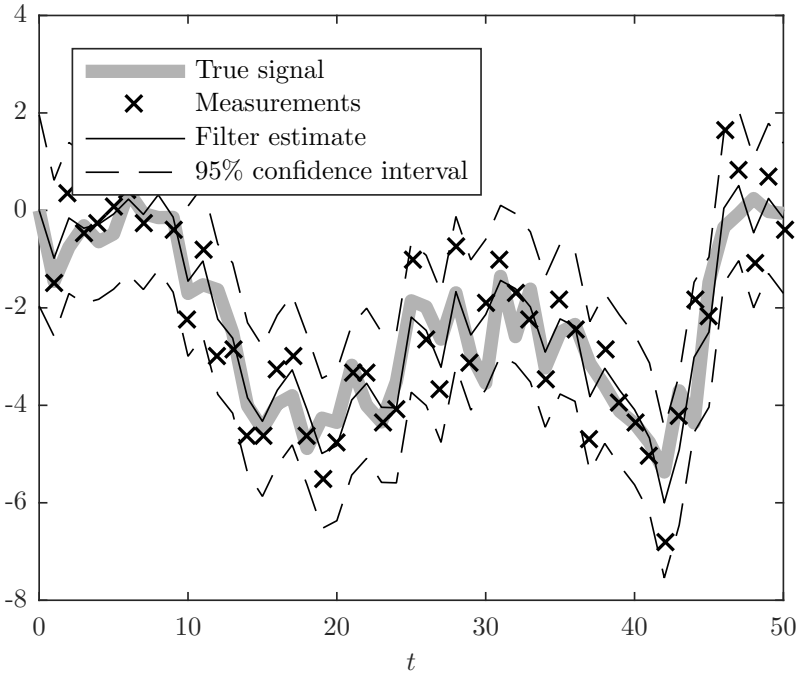

Figure 2.2 *Example of Kalman filter usage for random walk. At each point in time filter gives an optimal estimate for the data based on the made assumptions. Dashed lines represent the uncertainty of the filter estimate.*

In Figure 2.2 we show an example of using Kalman filter for a random walk. The filter estimates correspond rather well to the true signal, even though we have only noisy measurements available.

While Kalman filter is extremely useful tool, it assumes linearity in dynamic and measurement models. This is in many cases not reasonable assumption, as it isn't in our application of estimating the position and velocity of a satellite that is orbiting the Earth. In cases such as this, nonlinear filters have to be implemented.

2.3.2 Extended Kalman filter

Extended Kalman filter (EKF) is a solution to problem where dynamic model and measurement model are nonlinear. There are many versions of Extended Kalman filter, depending on model being continuous or discrete. We shall focus on the continuous-discrete version of the problem.

In general nonlinear state-space model, it is not possible to calculate the exact solution to the filtering problem. Therefore approximations are needed.

In Extended Kalman filter, Taylor series approximations of model functions are used at each timestep to calculate the next estimate. To form the approximations, we need the Jacobian matrix of the dynamic model function and measurement function with respect to state vector. In some cases, it is possible that this cannot be calculated. However, in our case, we can calculate the Jacobians and therefore, Extended Kalman filter suits our needs.

As we stated earlier, in Extended Kalman filter models are linearized and therefore they again lead to Gaussian distributions for the filtered states. Therefore, also with the Extended Kalman filter, we get an estimate for the mean and covariance matrix of the distribution.

Algorithm 2.2 (Continuous-discrete Extended Kalman filter) *If dynamic model function and measurement function are according to Definition 2.3, then we (analytically or numerically) solve differential equations*

$$\begin{aligned} \frac{d\mathbf{m}(t)}{dt} &= \mathbf{f}(\mathbf{m}(t), t), \\ \frac{d\Phi(t, t_{k-1})}{dt} &= \mathrm{F}_{\mathbf{x}}(t)\Phi(t, t_{k-1}) \end{aligned}$$

where $\mathrm{F}_{\mathbf{x}}$ *is the Jacobian matrix of* $\mathbf{f}$ *and the matrix* Φ *has property* $\Phi(t_{k-1}, t_{k-1}) = \mathrm{I}$. *The equations are integrated from* t_{k-1} *to* t_k *and if solutions to differential equations are* $\hat{\mathbf{m}}(t_k)$ *and* $\hat{\Phi}(t_k)$ *at point* t_k, *then predicted mean and covariance (a priori) are*

$$\begin{aligned} \mathbf{m}^-(t_k) &= \hat{\mathbf{m}}(t_k) \\ \mathrm{P}^-(t_k) &= \hat{\Phi}(t_k)\mathrm{P}(t_{k-1})\hat{\Phi}^T(t_k) + \mathrm{Q} \end{aligned}$$

and using measurement $\mathbf{y}_k$*, mean and covariance (a posteriori) are updated with*

$$\begin{aligned}
\boldsymbol{\mu}_k &= \mathbf{h}(\mathbf{m}^-(t_k)) \\
\mathrm{S}_k &= \mathrm{H}_{\mathbf{x}}\mathrm{P}^-(t_k)\mathrm{H}_{\mathbf{x}}^T + \mathrm{R}_k \\
\mathrm{K}_k &= \mathrm{P}^-(t_k)\mathrm{H}_{\mathbf{x}}^T\mathrm{S}_k^{-1}, \\
\mathbf{m}(t_k) &= \mathbf{m}^-(t_k) + \mathrm{K}_k(\mathbf{y}_k - \boldsymbol{\mu}_k), \\
\mathrm{P}(t_k) &= (\mathrm{I} - \mathrm{K}_k\mathrm{H}_{\mathbf{x}})\mathrm{P}^-(t_k)(\mathrm{I} - \mathrm{K}_k\mathrm{H}_{\mathbf{x}})^T - \mathrm{K}_k\mathrm{R}\mathrm{K}_k^T,
\end{aligned}$$

where $\mathrm{H}_{\mathbf{x}}$ *is the Jacobian of the measurement function* $\mathbf{h}$*.*

The algorithm is mostly the same presented and derived in [19, p. 269-278]. It is quite similar to Kalman filter in the discrete case (Algorithm 2.1) with two significant differences. First, the continuous model is transformed into discrete by integrating. Second, nonlinearities lead to approximations being used about dynamic and measurement functions. We also note that we use slightly different form of the covariance update that is, however, equivalent to the first one. This form is known as Joseph form and reason for using this is that while it is computationally a bit heavier, it has more stable numerical properties and since we compute filter values numerically, this form is more suitable [21].

Here we have written mean $\mathbf{m}$ and covariance P as a function of current time step to distinguish the continuous case from the discrete version. From now on, however, we shall use notation $\mathbf{m}_k := \mathbf{m}(t_k)$ and $\mathrm{P}_k := \mathrm{P}(t_k)$, even when we are referring to the continuous case.

This is a first-order version of the algorithm. This means that nonlinearities in the models are approximated to be linear. Higher-order versions of the algorithm also have been implemented for the cases, when the model is highly nonlinear, but we shall not consider them here.

2.3.3 Fixed-point smoother

With smoothing, we refer to a more general problem of estimating some state $\mathbf{x}_i$ based on a set of measurements $\mathbf{y}_{1:k}$, and $0 \leq i < k$. Generally, with smoothing we may estimate the whole state space $\mathbf{x}_{0:k}$. The algorithm, which corresponds to Kalman filter, the linear Gaussian case, is known as Rauch-Tung-Striebel smoother, which calculates the whole state space $\mathbf{x}_{0:k}$.

We may also be interested in some specific state. In this case using a general smoothing algorithm is not necessary since we are interested in only one state and we would perform lots of calculation for no reason. Instead of general smoothing,

we consider algorithms, which calculate the estimate of the state at some specific time t_i. This algorithm is known as fixed-point smoother.

In [29, p. 159-162], a general fixed-point smoother is presented. This is a generic method, which works with all filtering algorithms. We are interested in this case about continuous-discrete Extended Kalman filter, so we shall modify the equations to create a fixed-point smoother for this algorithm.

Algorithm 2.3 (Extended fixed-point smoother) *If the dynamic model and measurement model are according to Definition 2.3 and we are interested in state at time t_i based on measurements at times $t_1, \ldots, t_i, \ldots, t_n$, then we use equations of Algorithm 2.2 to obtain estimates $\mathbf{m}_k$ and P_k and*

- *if $k = i$, set $\mathrm{B}_k = \mathrm{I}$*
- *if $k > i$, update estimates $\mathbf{m}_k$ and P_k with*

$$\begin{aligned} \mathrm{G}_{k-1} &= \mathrm{P}_{k-1}\hat{\Phi}^T(\mathrm{P}_k^-)^{-1} \\ \mathrm{B}_k &= \mathrm{B}_{k-1}\mathrm{G}_{k-1} \\ \mathbf{m}_{i|k} &= \mathbf{m}_{i|k-1} + \mathrm{B}_k(\mathbf{m}_k - \mathbf{m}_k^-) \\ \mathrm{P}_{i|k} &= \mathrm{P}_{i|k-1} + \mathrm{B}_k(\mathrm{P}_k - \mathrm{P}_k^-)\mathrm{B}_k^T, \end{aligned}$$

where we have used notation $\mathbf{m}_{i|k} := \mathbf{m}(t_i)|\mathbf{y}_{1:k}$ and similarly for P.

This algorithm will mainly be what we will be using in this work when considering state estimation problems.

In this chapter we introduced some mathematical concepts from probability and measurement theory and also gave a few definitions for state-space models which were then used to introduce various filters. We did not introduce any deeper mathematical justification for the filters and for the derivation of the filters or more theory about filtering or stochastic processes in general, one may refer to [29, 19].

3. ORBIT PREDICTION

In this chapter, we will present a force model for the motion of a satellite. This can be used in both predicting the state of the satellite and also to form the dynamic model function for the Extended Kalman filter or smoother described in Chapter 2.

The equation of motion of the satellite is

$$\ddot{\mathbf{r}}(t) = \mathbf{a}(\mathbf{r}(t), t) = \sum_i \mathbf{a}_i(\mathbf{r}(t), t), \quad \mathbf{r}(t_0) = \mathbf{r}_0, \quad \dot{\mathbf{r}}(t_0) = \mathbf{v}_0, \tag{3.1}$$

where $\mathbf{a}_i$ is acceleration caused by the force i and $\mathbf{r}_0$ and $\mathbf{v}_0$ are position and velocity at the initial time t_0 obtained from the initial state estimation algorithm, which will be discussed more detailed in Section 3.4. By integrating (3.1), we can predict the position and velocity of a satellite at later time t.

We will consider the four largest forces acting on a satellite. These are the gravitation of the Earth, the Sun and the Moon and solar radiation pressure (SRP). The expressions for these are derived based on basic laws of physics and are mainly dependent only on time and the satellite's position. In this study, the forces which are derived this way are called analytical forces.

After the largest forces have been taken into account, we could continue by adding smaller terms in (3.1). Adding additional terms, for which expressions are derived based on physics of known forces, may not have a big effect on prediction accuracy [28]. Therefore we shall consider the problem also from the viewpoint of machine learning. This means we attempt to create a hybrid model, where mechanistic force model and data-driven approach are combined. Here, the latent forces that will be discussed in Chapter 4 come into play.

With the analytical and latent forces, (3.1) becomes

$$\ddot{\mathbf{r}} = \mathbf{a}_{\text{Earth}} + \mathbf{a}_{\text{Sun}} + \mathbf{a}_{\text{Moon}} + \mathbf{a}_{\text{SRP}} + \mathbf{a}_{\text{latent}}, \tag{3.2}$$

where $\mathbf{a}_{\text{Earth}}$, $\mathbf{a}_{\text{Sun}}$, $\mathbf{a}_{\text{Moon}}$ and $\mathbf{a}_{\text{SRP}}$ are the forces for which we will form explicit expressions and $\mathbf{a}_{\text{latent}}$ will be presented in Chapter 4. Here, we have left out explicit dependencies of position and time from the terms. As we discussed earlier, each

term is dependent on time and the satellite's position and term $\mathbf{a}_{\text{latent}}$ is additionally dependent on the velocity of the satellite.

First we take a look at the different coordinate systems required in orbit prediction and then consider what the expressions of analytical forces in (3.2) look like.

3.1 Coordinate systems

Satellite positioning gives rise to the need for different coordinate systems. Maintaining accurate systems is important, since positioning can require precision within meters or even centimeters and even the smallest rotations in coordinate systems lead to massive changes in satellite coordinates. We shall not focus too much on these coordinate systems and their complexities, we merely give an introduction to required systems and parameters related to them.

In positioning applications, we are often interested in coordinates with respect to the surface of the Earth. The Earth is rotating, so this leads to a rotating coordinate system. This kind of coordinate systems have a common generic name, *Conventional Terrestrial Reference System* (CTRS). This is an ideal coordinate system and different satellite constellations use different realizations. We shall use a common name for realizations, *Earth-Centered Earth-Fixed* (ECEF) coordinate systems.

We will also need coordinate systems that are not rotating. For example, when integrating the orbit of a satellite, we want to perform the integration in an inertial coordinate system. While it would be possible to formulate the equations of motion for rotating coordinate system, it is much simpler to use coordinate transformations and perform the integration in an inertial coordinate system. For our purposes, we fix the inertial coordinate system to the start of the prediction. More precisely, it is a realization of ECEF coordinate system at the time of our first observation. This concept of first observation will become clearer in Section 3.4, when we discuss about our initial state estimation algorithm.

For the coordinates of celestial bodies we also require another coordinate system. Neither of the above is suitable since we want an inertial coordinate system, which is consistent and not dependent on the time instant. One such system is *International Celestial Reference System* (ICRS). This coordinate system is bound to distant quasars and galaxies which are assumed to have no proper motion. This makes the coordinate system to not rotate with respect to time. The origin of the system is in the barycenter of the solar system, which means that technically the system does have some proper motion, but its effect is negligible.

In this context, another important concept is *Earth Orientation Parameters* (EOP).

The rotation axis of the Earth is not stable and changes in time. Therefore we will need some value for how much the current axis deviates from some standard. We shall not consider this problem too deeply, and shall use EOP provided by *International Earth Rotation and Reference Systems Service* (IERS). These are publicly available in [1].

Related to the EOP is also another parameter, which is usually considered together with EOP. This is called dUT1 and is related to different timescales. This parameter expresses the difference between Universal Time (UT1), which is defined to match the rotation of the Earth, and Coordinated Universal Time (UTC), which is time defined by a network of atomic clocks. In Universal Time, second is not constant, because the rotation of the Earth is slowly decelerating. The difference between the two (and so parameter dUT1) is always less than one second. This is achieved with leap seconds.

These parameters EOP and dUT1 become especially important in this context, since we aim to create an adaptive method, where we may continue for a long time with the same data, and therefore we need an estimate of EOP and dUT1 to perform coordinate transformations. EOP can be estimated together with the initial state of the satellite [8]. For dUT1, we shall not use any estimation methods and assume that this is delivered to the positioning device via some sort of assistance data.

We further introduce one coordinate system, which we will need in the context of latent force models and error analysis. This is defined by the position and velocity of the satellite and is known as *Radial-Transverse-Normal* (RTN) coordinate system. Radial direction is defined as direction from the center of the Earth to the satellite, normal direction is to the direction of the cross product of position and velocity of the satellite and transverse direction completes the right-handed coordinate system.

3.2 Analytical forces

The four largest forces are introduced briefly here. For more accurate description about the models and derivations of the formulas, one may refer to [23] or [31].

The gravitation of the Earth is by far the biggest force acting on a satellite. Because the satellite is relatively close to the Earth, a simple point mass model for the gravitation is not sufficient. We need to take into account the irregularities in the shape of the Earth.

The gravitation of the Earth can be modelled using Legendre polynomials. The

potential of the Earth's gravitational field is

$$U_{\mathrm{E}} = \frac{GM_{\mathrm{E}}}{r} \sum_{n=0}^{\infty} \sum_{m=0}^{n} \left[\left(\frac{R_{\mathrm{E}}}{r} \right)^n P_{nm}(\sin\phi) \Big(C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda) \Big) \right]. \quad (3.3)$$

Here r is a distance of the satellite from Earth's center, M_{E} and R_{E} are Earth's mass and radius, G is gravitational constant and λ and ϕ are the longitude and the latitude of the satellite. Terms P_{nm} are the associated Legendre polynomials of degree n and order m. Values for the coefficients C_{nm} and S_{nm} are from EGM2008 [25]. [1]

The potential of the Earth is calculated in ECEF, which is rotating with the Earth. In inertial coordinate system the acceleration caused by the potential in (3.3) is

$$\mathbf{a}_{\mathrm{Earth}} = \mathrm{R}_{\mathrm{ECEF}}^{-1} \nabla U_{\mathrm{E}}, \quad (3.4)$$

where $\mathrm{R}_{\mathrm{ECEF}}$ is the transformation matrix from inertial coordinate system to ECEF and ∇ denotes the gradient.

The Sun and the Moon are sufficiently far away from the satellite for their gravitation to be modelled as that of a point mass. Acceleration caused by the gravitation of a celestial body on the satellite relative to the motion of the Earth is

$$\mathbf{a}_{\mathrm{cb}} = GM_{\mathrm{cb}} \left(\frac{\mathbf{r}_{\mathrm{cb}} - \mathbf{r}_{\mathrm{Sat}}}{\|\mathbf{r}_{\mathrm{cb}} - \mathbf{r}_{\mathrm{Sat}}\|^3} - \frac{\mathbf{r}_{\mathrm{cb}}}{\|\mathbf{r}_{\mathrm{cb}}\|^3} \right), \quad (3.5)$$

where M_{cb} is the mass of the celestial body and $\mathbf{r}_{\mathrm{cb}}$ and $\mathbf{r}_{\mathrm{Sat}}$ are the positions of the celestial body and the satellite in an Earth-centered reference frame. Positions for celestial bodies can be obtained in a few ways and here we consider two that we have in use.

First possibility is to use analytical approximate expressions for the positions. Examples of these are derived in [23, p. 70-73]. Their use is quite simple and does not require any additional data. These are not as accurate as in some applications is required but usually in autonomous satellite predictions their accuracy level is sufficient.

Another solution is the use of almanac data. For example *Jet Propulsion Laboratory* (JPL) provides *Development Ephemerides* (DE) for positions of bodies of solar system in the form of Chebyshev polynomials [16]. In this work, we have used DE202 data, which are not the latest available ephemerides but they are compatible with the coordinate systems we currently have in use. These are much more accurate

[1] EGM stands for *Earth Gravitational Model*.

than analytical expressions and hence result in more accurate predictions. Downside is that almanac files are quite large and they would increase memory usage in the device.

We will not compare the prediction accuracy between the two approaches in this work.

Last force we will consider in our force model is solar radiation pressure. It is divided into two components, the first component describing the direct acceleration from the direction of the Sun and the other component describing the acceleration into an orthogonal direction from the original one. It is caused by the irregularities in the shape of the satellite and reflections of the solar radiation.

For direct solar radiation pressure we have

$$\mathbf{a}_{\text{SRP, direct}} = -\alpha_1 P_0 (1+\gamma) \frac{A_{\text{Sat}}}{m_{\text{Sat}}} \frac{\text{AU}^2}{\|\mathbf{r}_{\text{Sun}} - \mathbf{r}_{\text{Sat}}\|^2} \mathbf{e}_{\text{Sat,Sun}}, \tag{3.6}$$

where P_0 is the average solar radiation pressure at Earth's distance, γ is the radiation pressure coefficient, A_{Sat} and m_{Sat} are area and mass of the satellite and AU is the average distance of the Earth from the Sun [23]. Vectors $\mathbf{r}_{\text{Sun}}$ and $\mathbf{r}_{\text{Sat}}$ are the position of the Sun and the satellite and $\mathbf{e}_{\text{Sat,Sun}}$ is the unit vector from the satellite to Sun. Parameter α_1 is a coefficient which is estimated for each satellite individually. This way we do not need to take into account the different masses or areas of different satellite types but differences are taken into account in the parameter α_1.

If the satellite absorbed all the radiation from the Sun, the expression above would be enough. The satellite, however, reflects the radiation partially. This also causes a force for which we need a component orthogonal to the first one. This force is called *y-bias*. The y-direction is defined as

$$\mathbf{e}_{\text{y}} = \frac{\mathbf{r}_{\text{Sat}} \times (\mathbf{r}_{\text{Sun}} - \mathbf{r}_{\text{Sat}})}{\|\mathbf{r}_{\text{Sat}} \times (\mathbf{r}_{\text{Sun}} - \mathbf{r}_{\text{Sat}})\|}. \tag{3.7}$$

The acceleration in the y-direction is

$$\mathbf{a}_{\text{SRP, y-bias}} = 10^{-9} \cdot \alpha_2 \mathbf{e}_{\text{y}}, \tag{3.8}$$

where α_2 is again a parameter which is estimated for each satellite individually. The force is scaled with 10^{-9} so that parameters α_1 and α_2 are roughly the same magnitude to avoid numerical errors in the estimation phase.

Total acceleration caused by the solar radiation pressure is then

$$\mathbf{a}_{\text{SRP}} = \nu(\mathbf{a}_{\text{SRP, direct}} + \mathbf{a}_{\text{SRP, y-bias}}), \tag{3.9}$$

where $\nu \in [0, 1]$ is a parameter which models the shadows of celestial objects. The most considerable one is the shadow of the Earth, but in this work we also consider the shadow of the Moon. We use a conical model for the shadows of the bodies as described in [23, p. 80-83].

We shall take the shadow of the Moon into account only in cases, when the satellite is not affected by the shadow of the Earth. This means that if the satellite is fully or partially in Earth's shadow, the possible effect from Moon's shadow is neglected.

3.3 Estimation of solar radiation pressure parameters

We shall briefly consider the estimation of parameters α_1 and α_2 in equations (3.6) and (3.8). The process is explained in more detail in [9].

As we will explain in Section 4.3, additional parameters to be estimated can be augmented as a part of the state of a satellite. In this case, parameters α_1 and α_2 are considered to be constant. This means that the derivative is set to zero and small process noise is added.

In earlier work, these parameters have been estimated with use of large amounts of precise ephemeris data. In [9], parameters were estimated with one year of PE data, which was divided into 7-day arcs and parameters were estimated for each arc separately using filtering techniques from Chapter 2. Median of the filtered values was considered to be the correct value for the parameter. This is also the approach we will be using. These parameters can be calculated offline on a server, and then be sent to a positioning device as assistance data.

3.4 Initial state estimation of a satellite

The data from the broadcast ephemeris is not accurate enough to be used as an initial state in predictions. Specifically, the velocity estimate acquired directly from the broadcast message is too inaccurate. We wish to predict the position of the satellite for many days, or even weeks, which means that our initial state estimate needs to be accurate for our results to be reasonably accurate. We will solve this problem by fitting our force model to the orbit of the satellite with filtering methods presented in Chapter 2.

Because of nonlinearities, we shall use Extended Kalman filter. Moreover, because we are interested in the state of the satellite at some specific time instance, rather than the final time, we shall need the general fixed-point smoother.

More detailed description about the general idea of the initial state estimation algorithm used in this study can be found in [8].

The state of the satellite includes position and velocity in the inertial coordinate system. The dynamic model function can then be created using the force model constructed in this chapter and continuous-time model for the dynamics becomes

$$d\begin{bmatrix}\mathbf{r}(t)\\ \mathbf{v}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{v}(t)\\ \mathbf{a}(\mathbf{r}(t),t)\end{bmatrix} dt + \begin{bmatrix}0_{3\times 3}\\ \mathrm{R}_{\mathrm{RTN}}^{-1}(\mathbf{r}(t),\mathbf{v}(t))\end{bmatrix} d\beta, \tag{3.10}$$

where $\mathrm{R}_{\mathrm{RTN}}$ is a coordinate transformation matrix from inertial system to RTN-coordinate system and β describes Brownian motion which is also modelled in RTN-coordinate system. Unit vectors for this system are defined in Section 5.3. The noise in the model is included only in the velocity component as in [8]. Acceleration $\mathbf{a}$ can be calculated from (3.2). This equation corresponds to (2.2) and statistical inference can be done with methods presented in Chapter 2 to obtain the position and velocity of the satellite at the required time instant.

Data from broadcast ephemeris is extended into k position and velocity measurements. The time of ephemeris, or t_{toe} refers to the reference time of the ephemeris in question. For our purposes, we sample 3 hours of ephemeris data for estimation ($[t_{\mathrm{toe}} - 1.5\,\mathrm{h}, t_{\mathrm{toe}} + 1.5\,\mathrm{h}]$) with a 15-minute sampling interval, giving us 13 measurements.

The first sample of position and velocity are used to initialize the algorithm and the rest are inferred with Extended fixed-point smooother (Algorithm 2.3) using positions as measurements. This gives us the position and velocity at t_{toe}, which can then be used as initial conditions to differential equation (3.1).

4. LATENT FORCE MODELS

Orbit model presented in previous chapter gives a deterministic model for the motion of the satellite. This sort of mechanistic models usually work quite well in these types of situations. We could further try to improve the orbit prediction accuracy by adding more forces to the model. For example, we could consider gravity effects of other celestial bodies, consider ocean tide and solid tide effects, or use more precise model for SRP that would include reflection of radiation from Earth's surface, but this kind of approach does not significantly improve the prediction accuracy [28].

This leads us to consider a data-driven approach. We shall include additional forces, about which we have some prior knowledge, and which are then estimated more precisely based on data. This kind of forces are called latent forces and models including them are *latent force models* (LFM).

If the estimation were to work, a question then arises, could we estimate all forces using this data-driven approach? It is mentioned in [12] that using purely these kinds of models can fail when data is scarce relative to the complexity of the model or the model is forced to extrapolate outside its training regime. Needless to say, both these conditions take place when positioning device receives data for a short period of time and the orbit of a satellite should be predicted for days. This leads us to combine the mechanistic model and the data-driven approach to create a hybrid model.

In this chapter, we will introduce a concept of latent force models and then describe how these can be used in GNSS satellite orbit prediction.

4.1 About latent force models

Latent force models refer to force models where the exact source of the force is not known. Therefore, we do not have a physical explanation or knowledge about the origin of the force and hence constructing analytical expression for this is not possible.

Research on latent variables has been done for example by Titsias et al. in [35].

In their work, they considered Gaussian process models where latent variables were included. Their approach is more suitable for traditional machine learning problems, where large datasets are used to train the model. We need a method that can be used with small amount of data. They did not use latent forces in the sense that we will use them, but only latent variables in their model. They used marginal likelihoods, which we shall not consider in this study, to estimate values for their latent variables. This differs from our work in the sense that in our approach, we will sequentially update the values of latent forces.

Alvarez et al. considered latent force models in [10, 12]. In their work, they considered linear differential equations, and latent forces were added as non-parameteric Gaussian process models. Rather than using filtering approach that we will use, they analytically constructed expressions for the covariance matrices of the Gaussian processes, a method that is rather difficult to transform into a numerical method. They did, however, use a somewhat similar approach to ours, where mechanistic force model was created for a physical system, and then latent forces were added to create a hybrid model. The use of this kind of approach is beneficial, because even if we want accurate predictions, we do not need to create an overly complex mechanistic model but we can compensate the deficiencies by combining the physical model with a data-driven approach.

Särkkä et. al. have done work on sequential inference of latent force models [17, 30, 18], which means that latent forces are estimated with filters and smoothers such as those presented in Chapter 2. This is a simple numerical approach to estimating latent forces. From the perspective of this study, the most notable one is [18] where a latent force model is constructed for predicting the orbit of a satellite. This was, however, done based on precise data and only one test was made for one satellite. We will apply the described methods to also work in the context of autonomous positioning and test multiple satellites and constellations.

We also note here that even though we are talking about forces, we are actually modelling the accelerations. Since by Newton's second law $\mathbf{F} = m_{\text{Sat}}\mathbf{a}$, modelling either one of the two is equivalent. This, however, would become relevant if further analysis was to be made for example about the magnitudes of latent forces between two satellites, which are different type and hence may have different masses.

4.2 Latent forces in satellite orbit prediction

In Chapter 3 we presented our model for orbit prediction, which was based on integrating the orbit based on initial conditions and analytical expressions for forces from known sources. We of course cannot model every force and for forces that are not modelled with analytical expressions, we shall use a data-driven approach

instead. That is, we estimate the forces which have not yet been modelled using latent force models. The forces are estimated using the data that is received from the satellite. This approach was shown to improve orbit prediction accuracy in [18] in case of one prediction for one satellite.

We shall model the latent forces in RTN coordinate system, which was explained in Section 3.1. This coordinate system is based on the current position and velocity of the satellite in an inertial coordinate system. The latent forces need to be transformed to this same coordinate system to be used in prediction and state estimation. The forces are transformed to inertial coordinate system with

$$\mathbf{a}_{\text{latent}}(\mathbf{r}(t), \mathbf{v}(t), t) = \mathrm{R}_{\text{RTN}}^{-1}(\mathbf{r}(t), \mathbf{v}(t)) \begin{bmatrix} u_{\text{R}}(t) \\ u_{\text{T}}(t) \\ u_{\text{N}}(t) \end{bmatrix}, \tag{4.1}$$

where $\mathrm{R}_{\text{RTN}}(\mathbf{r}(t), \mathbf{v}(t))$ is the coordinate transformation matrix from inertial coordinate system to RTN coordinate system and $u_{\text{R}}(t)$, $u_{\text{T}}(t)$ and $u_{\text{N}}(t)$ are the latent forces in radial, transverse and normal direction, respectively. It should be noted that even though the total force is dependent on the position and the velocity of the satellite, the modelled force components u_{X} are only functions of time. This allows us to create much simpler models for them. We also note that we model the latent forces individually for each satellite. This means that we create a model that is valid for all satellites, but the latent forces are always specific to a particular satellite.

The three components are modelled independently as superpositions of stochastic resonators. There are mainly two reasons for this:

1. Satellites orbit the Earth and hence the majority of error components most likely consist of periodic components
2. Experimental evidence shows that in each direction, prediction errors exhibit oscillating behaviour. This is illustrated in Figure 4.1.

Stochastic resonators are characterized by differential equations of form

$$\frac{d^2 c_n(t)}{dt^2} = -(2\pi n f)^2 c_n(t) + w_n(t), \tag{4.2}$$

where c_n is the resonator in question, n is an integer, f is the base frequency of the oscillation and $w_n(t)$ is a noise term. In this work $w_n(t)$ is white noise with spectral density q_n. Essentially, (4.2) models sinusoidal oscillation with Gaussian noise. To be used in orbit prediction, we transform this model into state-space form. The process is further explained in [30].

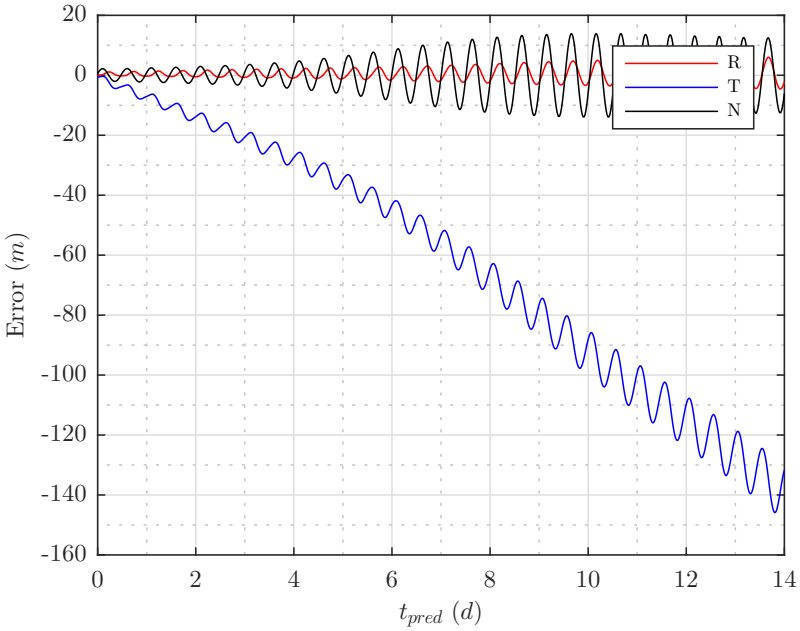

Figure 4.1 *Figure illustrating error components of a single two-week prediction for GPS satellite PRN 17. Prediction started from start of GPS week 1892. Here, the prediction was done using initial state estimation algorithm and analytical forces described in Chapter 3.*

Differential equation (4.2) can be written as a first-order differential equation in matrix form

$$\frac{d}{dt}\begin{bmatrix} c_n(t) \\ \dfrac{dc_n(t)}{dt} \end{bmatrix} = \mathrm{F}_n \begin{bmatrix} c_n(t) \\ \dfrac{dc_n(t)}{dt} \end{bmatrix} + \mathrm{L}_n w_n(t), \tag{4.3}$$

where matrices F_n and L_n are

$$\mathrm{F}_n = \begin{bmatrix} 0 & 1 \\ -(2\pi n f)^2 & 0 \end{bmatrix}, \qquad \mathrm{L}_n = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{4.4}$$

Notable here is that the form of (4.3) is compatible with the form of the equation which is used to model the state of the satellite in Section 3.4.

In [30], instead of the form in (4.3), the form for the matrix F_n actually used is

$$\mathrm{F}_n = \begin{bmatrix} 0 & 2\pi n f \\ -2\pi n f & 0 \end{bmatrix}. \tag{4.5}$$

In this case, the derivative term in the state and the noise term have to be scaled with $2\pi n f$. When frequency is considered to be constant, these two models are equivalent. This is the model we shall also use.

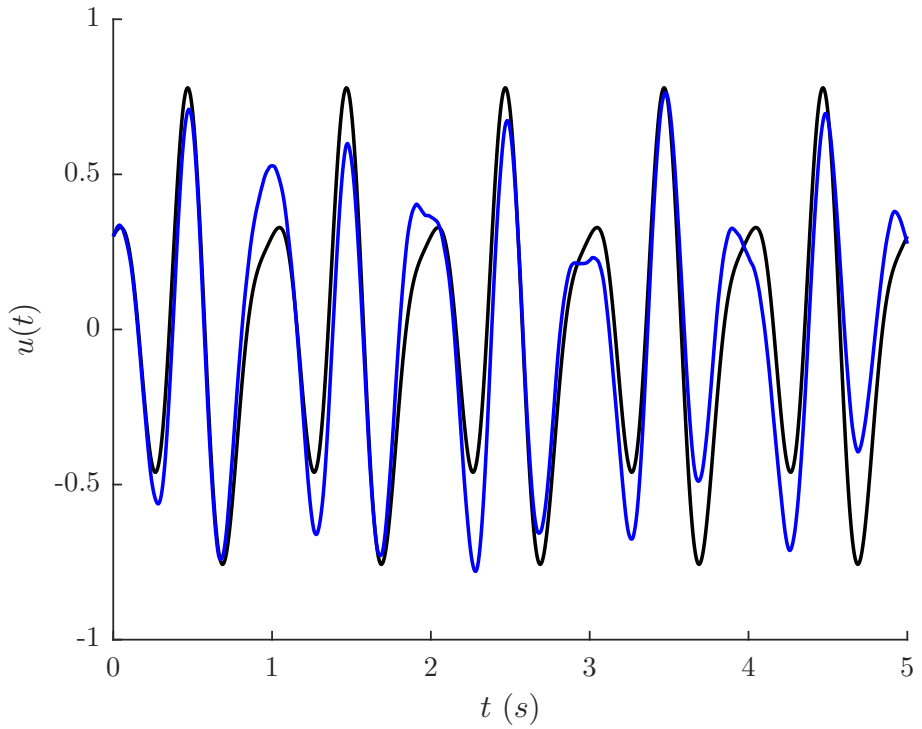

Figure 4.2 *Example of superposition of stochastic resonators simulated with a leapfrog method. Here three components are presented with base frequency $f = 1$. Black line indicates the mean trajectory, i.e. the solution without noise and blue line is one example with spectral density $q_n = 1$ for all components.*

Once the number of components N and harmonic frequency f have been decided, the form for the final matrices will be block-diagonal matrices of the components

$$\mathrm{F} = \mathrm{blkdiag}(\mathrm{F}_1, \mathrm{F}_2, \ldots, \mathrm{F}_N)_{2N\times 2N}, \qquad \mathrm{L} = \mathrm{blkdiag}(\mathrm{L}_1, \mathrm{L}_2, \ldots, \mathrm{L}_N)_{2N\times N}, \tag{4.6}$$

and for state and noise vectors for (4.3), we concatenate the components

$$\mathbf{c}(t) = \begin{bmatrix} c_1(t) & \frac{dc_1(t)}{dt} & \ldots & c_N(t) & \frac{dc_N(t)}{dt} \end{bmatrix}^T, \tag{4.7}$$

$$\mathbf{w}(t) = \begin{bmatrix} w_1(t) & \ldots & w_N(t) \end{bmatrix}^T. \tag{4.8}$$

We obtain a linear stochastic differential equation, which is compatible with our framework

$$\frac{d}{dt}\mathbf{c}(t) = \mathrm{F}\mathbf{c}(t) + \mathrm{L}\mathbf{w}(t). \tag{4.9}$$

Latent forces for (4.1) can then be calculated as a superposition of components $c_n(t)$

from equation

$$u(t) = \mathrm{H}\mathbf{c}(t) + b + \epsilon(t), \tag{4.10}$$

where H is a constant matrix that sums the components $c_1, \ldots, c_n$ from $\mathbf{c}$. We have also included additional terms to the model; b is bias of the latent force, since oscillation is not necessarily zero-centered, and term $\epsilon(t)$ to account for possible errors in the model.

Figure 4.2 shows an example of superposition of stochastic resonators $u(t)$, constructed according to (4.1)-(4.10). Trajectories were simulated with a leapfrog method described in [13]. We see that as more time passes, the errors compared to the mean trajectory become more significant.

Next we will consider how this model is combined with the satellite's dynamic state. This will allow the estimation of parameters in the model and therefore also estimation of the latent forces.

4.3 Augmentation of latent forces to initial state

Here we shall introduce a subscript $\mathrm{X} \in \{\mathrm{R}, \mathrm{T}, \mathrm{N}\}$ for equations and variables from the previous section. This corresponds to the latent force component in radial, transverse and normal directions.

The state-space model for the motion of the satellite is

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t) + \mathrm{D}(\mathbf{x}(t), t)\mathbf{w}_{\mathbf{x}}(t), \tag{4.11}$$

where $\mathbf{x}(t) = \begin{bmatrix}\mathbf{r}(t)^T & \mathbf{v}(t)^T\end{bmatrix}^T$ is the state of the satellite consisting of position and velocity, $\mathbf{f}(\mathbf{x}(t), t) = \begin{bmatrix}\mathbf{v}(t)^T & \mathbf{a}(\mathbf{r}(t), t)^T\end{bmatrix}^T$ is the dynamic model function, $\mathrm{D}(\mathbf{x}(t), t)$ is the dispersion matrix and $\mathbf{w}_{\mathbf{x}}(t)$ is the error term, which is considered to be white noise.

This model is used to estimate the initial state of the satellite. We already described the initial state estimation algorithm in Section 3.4 and now we will include latent force components to the algorithm.

As described in Section 4.2, a state-space model for the latent forces is

$$\begin{aligned}\frac{d}{dt}\mathbf{c}_{\mathrm{X}}(t) &= \mathrm{F}_{\mathrm{X}}\mathbf{c}_{\mathrm{X}}(t) + \mathrm{L}_{\mathrm{X}}\mathbf{w}_{\mathrm{X}}(t),\\ u_{\mathrm{X}}(t) &= \mathrm{H}_{\mathrm{X}}\mathbf{c}_{\mathrm{X}}(t) + b_{\mathrm{X}} + \epsilon_{\mathrm{X}}(t).\end{aligned} \tag{4.12}$$

The subscript $\mathrm{X} \in \{\mathrm{R}, \mathrm{T}, \mathrm{N}\}$ indicates the direction where the forces are modeled.

We now proceed to combine the model for the satellite and the model for the latent forces to create an augmented model.

Equations (4.11) and (4.12) can be combined to form an equation for the augmented state $\mathbf{x}_a(t)$ which consists of the original state, i.e. position and velocity of the satellite $\mathbf{x}(t)$, and the latent force components $\mathbf{c}_{\mathrm{X}}(t)$. Models are combined to describe the augmented state

$$\frac{d}{dt}\mathbf{x}_a(t) = \mathbf{f}_a(\mathbf{x}_a(t), t) + \mathrm{L}_a(\mathbf{x}_a(t), t)\mathbf{w}_a(t). \tag{4.13}$$

Here augmented state $\mathbf{x}_a(t)$ and augmented model function $\mathbf{f}_a(\mathbf{x}_a(t), t)$ are

$$\mathbf{x}_a(t) = \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{c}_{\mathrm{R}}(t) \\ \mathbf{c}_{\mathrm{T}}(t) \\ \mathbf{c}_{\mathrm{N}}(t) \\ \mathbf{b} \end{bmatrix}, \quad \mathbf{f}_a(\mathbf{x}_a(t), t) = \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{a}_a(\mathbf{r}(t), \mathbf{v}(t), t) \\ \mathrm{F}_{\mathrm{R}}\mathbf{c}_{\mathrm{R}}(t) \\ \mathrm{F}_{\mathrm{T}}\mathbf{c}_{\mathrm{T}}(t) \\ \mathrm{F}_{\mathrm{N}}\mathbf{c}_{\mathrm{N}}(t) \\ \mathbf{0}_{3\times 1} \end{bmatrix} \tag{4.14}$$

where $\mathbf{b} = \begin{bmatrix} b_{\mathrm{R}} & b_{\mathrm{T}} & b_{\mathrm{N}} \end{bmatrix}^T$ is the constant bias vector which is estimated together with the state in a similar way that SRP parameters are estimated (Section 3.3). The dispersion matrix $\mathrm{L}_a(\mathbf{x}_a(t), t)$ is

$$\mathrm{L}_a = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times N_{\mathrm{R}}} & \mathbf{0}_{3\times N_{\mathrm{T}}} & \mathbf{0}_{3\times N_{\mathrm{N}}} & \mathbf{0}_{3\times 3} \\ \mathrm{R}^{-1}_{\mathrm{RTN}}(\mathbf{r}(t), \mathbf{v}(t)) & \mathbf{0}_{3\times N_{\mathrm{R}}} & \mathbf{0}_{3\times N_{\mathrm{T}}} & \mathbf{0}_{3\times N_{\mathrm{N}}} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{2N_{\mathrm{R}}\times 3} & \mathrm{L}_{\mathrm{R}} & \mathbf{0}_{2N_{\mathrm{R}}\times N_{\mathrm{T}}} & \mathbf{0}_{2N_{\mathrm{R}}\times N_{\mathrm{N}}} & \mathbf{0}_{2N_{\mathrm{R}}\times 3} \\ \mathbf{0}_{2N_{\mathrm{T}}\times 3} & \mathbf{0}_{2N_{\mathrm{T}}\times N_{\mathrm{R}}} & \mathrm{L}_{\mathrm{T}} & \mathbf{0}_{2N_{\mathrm{T}}\times N_{\mathrm{N}}} & \mathbf{0}_{2N_{\mathrm{T}}\times 3} \\ \mathbf{0}_{2N_{\mathrm{N}}\times 3} & \mathbf{0}_{2N_{\mathrm{N}}\times N_{\mathrm{R}}} & \mathbf{0}_{2N_{\mathrm{N}}\times N_{\mathrm{T}}} & \mathrm{L}_{\mathrm{N}} & \mathbf{0}_{2N_{\mathrm{N}}\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times N_{\mathrm{R}}} & \mathbf{0}_{3\times N_{\mathrm{T}}} & \mathbf{0}_{3\times N_{\mathrm{N}}} & \mathrm{I}_{3\times 3} \end{bmatrix}, \tag{4.15}$$

and the augmented noise term $\mathbf{w}_a(t)$

$$\mathbf{w}_a(t) = \begin{bmatrix} \mathbf{w}_{\mathbf{x}}(t) \\ \mathbf{w}_{\mathrm{R}}(t) \\ \mathbf{w}_{\mathrm{T}}(t) \\ \mathbf{w}_{\mathrm{N}}(t) \\ \mathbf{w}_{\mathrm{b}} \end{bmatrix}, \tag{4.16}$$

where $\mathbf{w}_{\mathrm{b}}$ is a small noise term to allow small variations in the bias with respect to time. We are therefore not strictly speaking modelling a constant but another time-dependent variable. In many applications this is a suitable approach as justified in [29, p. 185-186].

When creating the model we have also used a simplifying assumption that latent force components are independent. For our needs, this assumption is reasonable, and we leave creating a more complex model as a topic for further research.

Equation (4.13) can be solved with similar methods as the initial state problem presented in Section 3.4. Calculation of Jacobians for the covariance matrix propagation is easy since derivatives of latent force components appear linearly in $\mathbf{f}_a$.

It should be noted that the acceleration $\mathbf{a}_a(\mathbf{r}(t), \mathbf{v}(t), t)$ is now dependent on both the position and velocity of the satellite. This is because the direction of the latent forces is dependent on the RTN coordinate system and therefore the velocity of the satellite. Furthermore, we have made the distinction between the two accelerations $\mathbf{a}(\mathbf{r}(t), t)$ and $\mathbf{a}_a(\mathbf{r}(t), \mathbf{v}(t), t)$ by introducing a subscript. The latter one includes the latent forces and the former one does not. These are connected by

$$\mathbf{a}_a(\mathbf{r}(t), \mathbf{v}(t), t) = \mathbf{a}(\mathbf{r}(t), t) + \mathbf{a}_{\text{latent}}(\mathbf{r}(t), \mathbf{v}(t), t). \tag{4.17}$$

We calculate $\mathbf{a}_{\text{latent}}(\mathbf{r}(t), \mathbf{v}(t), t)$ from (4.1). The term $\mathbf{a}(\mathbf{r}(t), t)$ is as explained in Chapter 3.

We have gathered all vectors and matrices used in latent force model and augmented model in Appendix A.

5. ERROR ANALYSIS IN GNSS POSITIONING

To compare the accuracy of the predictions, we need to consider how the prediction error should be calculated. We are interested in error relevant to positioning, and since pseudorange error caused by the error in position is basically only error in the direction from the satellite to positioning device, simple three-dimensional error is not a good error measure.

First we take a look at different satellite constellations that we will be included in our analysis and then we present how we will compare the prediction errors.

5.1 GNSS constellations

In this work we consider a few different GNSS. Naturally, we will focus on GPS but also we shall consider Russian satellite system GLONASS and Chinese Beidou. There are also other constellations, for example European Galileo satellite system and, while not global but regional systems, Japanese QZSS (*Quasi-Zenith Satellite System*) and Indian IRNSS (*Indian Regional Navigational Satellite System*), but we shall not consider results for these systems in this work. For general information about principles of positioning can be found in [22] (mostly from viewpoint of GPS) and information about different GNSS constellations in [20, 26].

We distinguish the satellites from each other with a *Pseudo-random number* (PRN). In this work, we consider this to be an identification tag for a satellite, so that each satellite is uniquely specified by constellation and PRN.

5.1.1 GPS

Global Positioning System (GPS) is a navigation satellite system developed by the United States. A plan to develop a satellite-based navigation system was proposed already in the 1960s and first plan was approved in 1973. At this point the application was purely for military purposes. First actual launch was in 1978, when Navstar 1 -satellite was launched into orbit. The development of the system delayed and complete constellation was ready in 1995.

The number of GPS satellites is at least 24 at all times, which is the minimum number required by the definition of the system. Satellites have been divided into six orbits equally spaced around the Earth and with 55° inclination. The total number of operational satellites through the years has been $30-32$. Extra satellites are meant to improve the reliability and accuracy of the system. Also, this allows broken satellites to quickly be replaced with new ones. GPS orbit period is exactly one half of the sidereal day of the Earth. Because the Earth is also rotating, this means that satellite is above the same point of the surface every day at the same time. [26]

The full broadcast ephemeris transmitted by a GPS satellite has by default a fit interval of four hours [6]. Broadcast data is valid only during this period, meaning $t_{\text{toe}} \pm 2$ hours. Data outside this time window is unreliable and should not be used.

5.1.2 GLONASS

Globalnaya Navigatsionnaya Sputnikovaya Sistema, commonly known as GLONASS, is a satellite system originally developed by the Soviet Union and nowadays maintained by Russia. Also originally designed for military purposes, GLONASS was started in 1982, and in 1996 it was in planned scope.

GLONASS system includes 24 satellites. They are, however, divided into three orbits with 64.8° inclination. This makes GLONASS work a bit better compared to GPS in the polar regions of the Earth. The orbit of GLONASS is, from theoretical standpoint of celestial mechanics a bit more stable than GPS. This is due to the fact that GPS satellites orbit the Earth exactly twice a day. This causes minor disruptions to magnify over time. With GLONASS, satellite is above the same point on Earth every 8 days. The constellation, however, "looks" similar every day, because every orbit has 8 satellites, and after one day, another satellite is where one was a day earlier. GLONASS satellites need fewer maneuvers than GPS satellites, because the orbit period is not as synchronized with Earth's rotation. [26]

One particular problem with GLONASS, when considering autonomous predictions, is the short validity time of GLONASS broadcast ephemerides. According to [4], GLONASS parameters are updated every 15 minutes and one broadcast is valid for only $t_{\text{toe}} \pm 15$ minutes. As a result, we do not get enough data points for filtering the initial state of a GLONASS satellite. To fix this problem, we use by default two broadcast ephemerides in initial state estimation of a GLONASS satellite. The method is further explained in [36].

5.1.3 Beidou

Beidou Weixing Daohang Xitong, or simply Beidou, is a satellite system of the Chinese Peoples Republic. First started in 2000, it was taken into use in China in 2012 and is planned to be released for global use in 2020. The constellation earlier went under the name Compass.

Beidou is different from other systems in regard of its satellite orbits; Beidou has three different kinds of orbits for its satellites. Most of the satellites in the final constellation will be *Medium-Earth Orbit* (MEO) satellites, which are similar to GPS and GLONASS satellites. However, Beidou also has *Geostationary Orbit* (GEO) and *Inclined Geosynchronous Orbit* (IGSO), whose orbital period matches rotation of the Earth. This means that some of the satellites are above the same region all the time. This is why Beidou has optimal performance in parts of China, Oceania and Far East. [26]

The different orbit types bring their own challenges to autonomous prediction. All methods need to be tested to be working for each different orbit type. We will view the results for each orbit type separately in Chapter 6.

5.2 Error measures

We will consider the errors in Radial-Transverse-Normal coordinate system. The simplest approach would be to calculate the three-dimensional error of the prediction, but since different directions contribute differently to positioning error, we shall take a different approach. Instead, we shall consider how different directions contribute to positioning error on average on when the positioning device is on Earth's surface.

For accuracy of the broadcast ephemeris, SISRE (*Signal-In-Space-Range-Error*) is widely used as error metric. Since our goal is to mimic the behaviour of broadcast ephemerides, we will also choose this as metric for our analysis. SISRE describes the average positioning error caused by the different error components. By far, the largest contribution comes from the radial direction but when satellite is close to horizon when viewed from Earth's surface, transverse and normal directions also become significant.

It should be noted that usually the error consists of the clock and position error [5]. In this work we are only interested in position error and therefore we omit the clock error components from the following formulas. It is, however, quite straightforward to include the clock error in the calculations.

In literature, the term SISRE is used mainly in two different ways. Here we shall discuss the two approaches.

SISRE can be related to a single prediction, where weights are given to radial, transverse and normal components. SISRE is then calculated with

$$SISRE_{\text{single}} = \sqrt{w_R^2 \Delta\text{R}^2 + w_{T,N}^2(\Delta\text{T}^2 + \Delta\text{N}^2)}, \tag{5.1}$$

where ΔR, ΔT and ΔN are error components in radial, transverse and normal direction, respectively. Weights w_{R} and $w_{\text{T,N}}$ depend on the altitude of the satellite. The weights are squared because when the clock is included, the formula is in a bit different form, where w_{R} is not squared. Values for these weights for different constellations are gathered in Table 5.1.

	GPS	GLONASS	Beidou MEO	Beidou IGSO/GEO
w_{R}	0.98	0.98	0.98	0.99
$w_{\text{T,N}}^2$	$\frac{1}{49}$	$\frac{1}{45}$	$\frac{1}{54}$	$\frac{1}{126}$

Table 5.1 *SISRE coefficient values for different constellations. Values are from [24]. Note that one parameter is squared while the other is not.*

The idea for derivation of this formula for Beidou has been presented in [14]. This is, however, a general method and may be applied to any constellation. We shall also derive the formula and expressions for weights w_{R} and $w_{\text{T,N}}$ in Section 5.3 based on [14].

In this case SISRE becomes a measure for a single error of one satellite. Multiple errors then produce multiple SISRE values and statistical analysis can be done for these. This approach is used in [14, 5].

Another use of SISRE is a statistical approach, where SISRE is used to describe the statistics of collection of error components from multiple errors. This is the approach used in [24]. Then, root mean square (rms) of error components is used to calculate the statistical meaning of SISRE with

$$SISRE_{\text{statistical}} = \sqrt{w_R^2\,\text{rms}(\Delta\text{R})^2 + w_{T,N}^2(\text{rms}(\Delta\text{T})^2 + \text{rms}(\Delta\text{N})^2)}, \tag{5.2}$$

which is quite similar to (5.1). Now the term SISRE refers to a single value for a large set of errors, which then can be used as performance measure. Sometimes instead of rms, other values, for example standard deviation, σ (or variance σ^2) is used.

The two definitions are connected with

$$\begin{aligned}
\mathrm{rms}(SISRE_{\mathrm{single}}) &= \sqrt{\frac{\sum SISRE_{\mathrm{single}}^2}{n}} = \sqrt{\frac{\sum w_{\mathrm{R}}^2 \Delta\mathrm{R}^2 + w_{\mathrm{T,N}}^2(\Delta\mathrm{T}^2 + \Delta\mathrm{N}^2)}{n}} \\
&= \sqrt{w_{\mathrm{R}}^2 \frac{\sum \Delta\mathrm{R}^2}{n} + w_{\mathrm{T,N}}^2 \left(\frac{\sum \Delta\mathrm{T}^2}{n} + \frac{\sum \Delta\mathrm{N}^2}{n} \right)} \\
&= \sqrt{w_{\mathrm{R}}^2 \,\mathrm{rms}(\Delta\mathrm{R})^2 + w_{\mathrm{T,N}}^2 \Big(\mathrm{rms}(\Delta\mathrm{T})^2 + \mathrm{rms}(\Delta\mathrm{N})^2 \Big)} \\
&= SISRE_{\mathrm{statistical}}.
\end{aligned}$$

Therefore the two are tightly connected and yield similar results. Care should be used in the terminology, since SISRE can refer to single prediction but also to rms or some other statistical measure.

In [15], one more meaning for SISRE is used. Here, SISRE means the range error in worst-user location (commonly known as *wul*). Although the abnormal use of the term is explicitly specified, it further illustrates how the term SISRE is used in different meanings.

In this work, we shall use SISRE to mean the rms pseudorange error from each location on the surface of the Earth, where the satellite in question is visible, caused by the error in satellite position. Therefore we calculate SISRE with (5.1) and analyze a set of SISRE values. We choose this approach for a specific reason.

Generally, SISRE is used as an accuracy estimate for BE data. Since the final goal in our study is to mimic the behaviour of BE data, naturally this is a good error measure. However, sometimes our prediction algorithm fails and produces outliers, which do not characterize the general behaviour of the data. This also causes measures such as rms or variance not to be the appropriate error measure.

We consider a bit different approach than generally is used. We shall use definition of SISRE from (5.1) to calculate error for each individual prediction. Then, instead of rms error, we shall use error quantiles. Since (5.1) always produces positive values, this approach should not cause problems.

We shall focus mainly on 68% and 95% quantiles. This would roughly correspond to 1σ and 2σ, if the errors were normally distributed. It should also be noted that we assume our error component distributions to be zero-centered and therefore, if the errors were normally distributed, our 68% quantile would match the definition of SISRE in (5.2).

5.3 Derivation of the SISRE formula

We will start from the RTN coordinate system, with definitions of unit vectors

$$\mathbf{r}_{\mathrm{R}} = \frac{\mathbf{r}}{\|\mathbf{r}\|}, \tag{5.3a}$$

$$\mathbf{r}_{\mathrm{N}} = \frac{\mathbf{r} \times \mathbf{v}}{\|\mathbf{r} \times \mathbf{v}\|}, \tag{5.3b}$$

$$\mathbf{r}_{\mathrm{T}} = \mathbf{r}_{\mathrm{N}} \times \mathbf{r}_{\mathrm{R}}, \tag{5.3c}$$

where $\mathbf{r}$ and $\mathbf{v}$ are the position and velocity of the satellite in inertial coordinate system. Vectors $\mathbf{r}_{\mathrm{R}}$, $\mathbf{r}_{\mathrm{N}}$ and $\mathbf{r}_{\mathrm{T}}$ are radial, transverse and normal direction unit vectors and define our coordinate axis. The vectors in this coordinate system are of form $[\mathrm{T}, \mathrm{N}, \mathrm{R}]^T$. This order is chosen, so that we can easily use familiar spherical coordinate transformations. We place the origin to the center of the Earth.

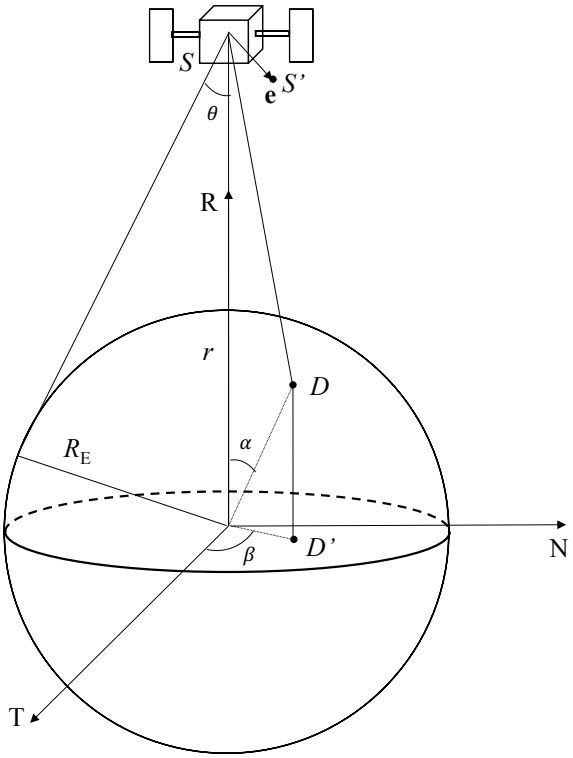

Figure 5.1 *Figure (not to scale) illustrating the RTN coordinate system and required variables.*

In Figure 5.1 we present the coordinate system and required variables. R_{E} is the radius of the Earth and r is distance of the satellite from Earth's center. Angles α and β are elevation and azimuth angles. Point D represents random point on the

surface of the Earth visible from the satellite and D' is projection of point D to TN-plane. Point S is the location of the satellite and point S' is predicted position of the satellite. Vector $\mathbf{e}$ is the prediction error vector ($\mathbf{e} = \overline{SS'} = [\Delta\mathrm{T}, \Delta\mathrm{N}, \Delta\mathrm{R}]^T$).

We first define vector $\mathbf{l}$ as unit vector to the direction of $\overline{SD}$. Using trigonometry from Figure 5.1, in terms of angles α and β, we get

$$\mathbf{l} = \frac{\overline{SD}}{\|\overline{SD}\|} = \frac{[R_\mathrm{E}\sin\alpha\cos\beta, R_\mathrm{E}\sin\alpha\sin\beta, R_\mathrm{E}\cos\alpha - r]^T}{\sqrt{R_\mathrm{E}^2 - 2R_\mathrm{E}r\cos\alpha + r^2}}. \tag{5.4}$$

The error in pseudorange measurement is given by the difference in the norms

$$\Delta\rho = \|\overline{SD}\| - \|\overline{S'D}\|. \tag{5.5}$$

When $\|\mathbf{e}\|$ is small compared to $\|\overline{SD}\|$, as it is in our case, we may approximate the pseudorange error using vector projection

$$\Delta\rho \approx \mathbf{e}\cdot\mathbf{l}. \tag{5.6}$$

We want to describe the average contribution of different error components to $\Delta\rho$ when point D is visible from the satellite. Area A is the area on the surface of the Earth that is visible from the satellite. As measure we shall use root mean square of pseudorange error. Now, instead of multiple error measurements as in statistical definition of SISRE, we use mean of different pseudorange errors due to different locations on the surface of the Earth. We are also interested in a continuous surface on the Earth, so we shall use continuous version of rms. This gives us the SISRE formula

$$SISRE = \mathrm{rms}(\Delta\rho) = \sqrt{\iint_A [\mathbf{e}\cdot\mathbf{l}]^2 \frac{dA}{\iint_A dA}}, \tag{5.7}$$

where the integral is normalized for mean value.

For infinitesimal changes in α and β, we get infinitesimal area on the surface of the Earth to be $dA = R_\mathrm{E}^2\sin\alpha d\alpha d\beta$. Variable α is integrated from 0 to $\frac{\pi}{2} - \theta$ and β is integrated from 0 to 2π. Using these, normalization constant becomes

$$\begin{aligned}\iint_A dA &= \int_0^{2\pi}\int_0^{\frac{\pi}{2}-\theta} R_\mathrm{E}^2\sin\alpha d\alpha d\beta \\ &= 2\pi R_\mathrm{E}^2(-\cos(\frac{\pi}{2} - \theta) + \cos 0) = 2\pi R_\mathrm{E}^2(1 - \sin\theta).\end{aligned} \tag{5.8}$$

Combining (5.7) and (5.8) then gives

$$SISRE = \sqrt{\int_0^{2\pi} \int_0^{\frac{\pi}{2}-\theta} [\mathbf{e} \cdot \mathbf{l}]^2 \frac{\sin\alpha}{2\pi(1-\sin\theta)} d\alpha d\beta}. \tag{5.9}$$

Next we shall consider the term $[\mathbf{e}\cdot\mathbf{l}]^2$ in terms of α and β. Combining the definition of $\mathbf{e}$ and (5.4) gives

$$\begin{aligned}[\mathbf{e} \cdot \mathbf{l}]^2 &= \frac{(\Delta \mathrm{T} R_\mathrm{E} \sin\alpha\cos\beta + \Delta \mathrm{N} R_\mathrm{E} \sin\alpha\sin\beta + \Delta \mathrm{R}(R_\mathrm{E}\cos\alpha - r))^2}{R_\mathrm{E}^2 - 2R_\mathrm{E} r\cos\alpha + r^2} \\ &= \frac{1}{R_\mathrm{E}^2 - 2R_\mathrm{E} r\cos\alpha + r^2}(\Delta \mathrm{T}^2 R_\mathrm{E}^2 \sin^2\alpha\cos^2\beta + \Delta \mathrm{N}^2 R_\mathrm{E}^2 \sin^2\alpha\sin^2\beta \\ &+ \Delta \mathrm{R}^2 (R_\mathrm{E}\cos\alpha - r)^2 + E).\end{aligned} \tag{5.10}$$

In terms of β, term E in above includes only terms with $\cos\beta$, $\sin\beta$ or $\sin\beta\cos\beta$. When these are integrated in (5.9) from 0 to 2π, they equal zero and hence disappear when integrated. In addition, remaining terms in the integral are separable and since

$$\int_0^{2\pi} \cos^2\beta d\beta = \int_0^{2\pi} \sin^2\beta d\beta = \pi,$$

double integrals can easily be reduced to single integrals. Now, introducing the notation

$$w_{\mathrm{T,N}}^2 = \frac{R_\mathrm{E}^2}{2(1-\sin\theta)} \int_0^{\frac{\pi}{2}-\theta} \frac{\sin^3\alpha}{R_\mathrm{E}^2 - 2R_\mathrm{E} r\cos\alpha + r^2} d\alpha, \tag{5.11a}$$

$$w_\mathrm{R}^2 = \frac{1}{1-\sin\theta} \int_0^{\frac{\pi}{2}-\theta} \frac{\sin\alpha(R_\mathrm{E}\cos\alpha - r)^2}{R_\mathrm{E}^2 - 2R_\mathrm{E} r\cos\alpha + r^2} d\alpha, \tag{5.11b}$$

and using the results above, we get the familiar formula

$$SISRE = \sqrt{w_\mathrm{R}^2 \Delta \mathrm{R}^2 + w_{\mathrm{T,N}}^2 (\Delta \mathrm{T}^2 + \Delta \mathrm{N}^2)}. \tag{5.12}$$

Parameter θ is calculated from the altitude of the satellite with $\sin\theta = \frac{R_\mathrm{E}}{r}$ as in Figure 5.1. Integrals in (5.11) can be calculated numerically to get weights w_R and $w_{\mathrm{T,N}}$. For example, substituting values for GPS ($R_\mathrm{E} = 6.37 \cdot 10^6$m, $r = 2.66 \cdot 10^7$m), we get values $w_\mathrm{R} = 0.979$ and $w_{\mathrm{T,N}}^2 = 0.0203 \approx \frac{1}{49}$, which correspond to values in Table 5.1.

6. RESULTS FOR ORBIT PREDICTION

In this chapter we will take a look at the performance of satellite orbit prediction using latent force models. We shall see that use of LFM greatly improves the accuracy of the predictions in many cases.

We will explain how the parameters for the model are chosen. Also, we shall consider adaptive approach, where new broadcasts are used to estimate the latent forces and create a testing setup corresponding to real-life scenario.

We will test our model with different satellite constellations and see that our model works with all constellations and we shall present the results with GPS, GLONASS and Beidou. The results for every satellite will not be presented here. Rather we show the behaviour of a few satellites that illustrate the characteristic behaviour of a typical constellation satellite. In Appendix B we have collected results for each satellite of each constellation at a single prediction length.

6.1 Choice of parameters

The model has a few free parameters. We shall take a look at how the parameters were chosen.

First we have the base frequency of the oscillator f. It is quite natural to choose this so that it corresponds to one orbital period. Errors in RTN-directions usually follow this oscillation frequency. This is illustrated in Figure 4.1, where a single prediction is done for a GPS satellite without latent force model.

This way f can be calculated as a reciprocal of the orbital period

$$f = \frac{1}{T_{orb}} = \frac{1}{a \cdot 86400\,\mathrm{s}}, \tag{6.1}$$

where a is the ratio of the orbital period and one solar day. Values of a for different satellite types are summarized in Table 6.1.

Next, we have the number of harmonic components in each direction. In this work, we chose the same number for every direction, $N = 3$. In [18], the number of

GPS	GLONASS	Beidou MEO	Beidou IGSO/GEO
0.4986	0.4688	0.5368	0.9972

Table 6.1 *Values of parameter a for different satellite systems. Durations of orbital periods are found in [3].*

components was 7 for radial and transverse direction and 10 for normal direction. Our main interest is the use of the algorithm in practical application, where the computation time also has to be minimized. Also, we use data from the broadcast ephemerides, which is not as accurate as precise data. We tried different values of N, and prediction accuracy wasn't practically affected with increasing N. Increasing the number of components, however, rapidly increases the computation time, which we will discuss further in Section 6.6.

The last parameter is spectral density of the noise terms q_n. Again for simplicity, we choose the same parameter for all components and all directions. This is because we are trying to create a general model which works with all constellations and all satellites, and approximate values are sufficient to model the correct magnitude of the noise. By choosing a parameter value to be the same for all constellations, we save memory and these parameters don't have to calculated separately for every satellite, which would be computationally quite expensive.

We assume that the noise level is some percentage units of the amplitude of the latent forces. With this assumption, we model the forces as near-sinusoidal, but the phase and amplitude are also subject to change. In [23, p. 55], a figure is presented with the magnitude of accelerations caused by the major forces. The level of acceleration caused by the forces that are not included in our model is roughly $10^{-9}\,\frac{\mathrm{m}}{\mathrm{s}^2}$. Same conclusion was given in [28].

We choose the noise level to correspond to our integration step size in EKF, which is 900 seconds. This gives us a discrete model, where the covariance matrix for each stochastic resonator component in each direction is

$$\mathrm{Q}_{\text{latent}} = \begin{bmatrix} 0 & 0 \\ 0 & q^2 \end{bmatrix}. \tag{6.2}$$

We choose $q = 10^{-11}\,\frac{\mathrm{m}}{\mathrm{s}^2}$ in this model. This is the noise for each resonator. For estimating the bias parameter b_{X}, we chose variance to be $q = 10^{-12}\,\frac{\mathrm{m}}{\mathrm{s}^2}$ for each direction. These noise levels satisfy our requirements described above.

6.2 Testing setup

We shall simulate the predictions with BE and PE data. The broadcast data is in RINEX-format (*Receiver Independent Exchange*) [7], while the precise data is given in SP3-format [33], which means as coordinates in ECEF with 15-minute sampling interval. The data is provided by *International GNSS Services* (IGS) and more specifically *Multi-GNSS Experiment* (MGEX). These are publicly available for download and more information can be found in [2].

Normally prediction accuracy has been tested by performing separate independent predictions of certain length from broadcast ephemerides and then combine the results. Since we want to estimate additional variables and vary the number of ephemerides and times between consecutive ephemerides, this sort of test scenario is not sufficient and hence we need to create a new testing setup.

In the test scenario, we want to replicate a real-life situation, where broadcasts are received somewhat randomly and the time difference between consecutive broadcasts can vary. This simulates a device that is turned on every now and then and then it receives new data.

We shall choose a maximum interval between two consecutive broadcast ephemerides. Starting from some point in time we receive a BE and then we randomly select the next BE from the interval. The state estimate and the covariance matrix are then integrated to this time instant, and measurements are used to update the smoother estimates and the procedure is repeated for certain amount of time.

Because we always choose the next ephemeris randomly, we get different prediction lengths. We gather these and measure the performance as a function of prediction time. If the prediction gets close to maximum interval length, we have fewer predictions and therefore accuracy during the final day is always cut off in the following figures.

From every received broadcast also a single prediction is made using only the data of that broadcast without any use of latent forces. The error is calculated by comparing the predicted position to precise data. We calculate the SISRE during the prediction for both approaches. Prediction accuracies using both methods are then compared with 68% and 95% quantiles.

We also set a threshold for the error, that if this is exceeded, the filter is reset. This is to prevent the case, that previous states do not affect the estimates, when satellite is maneuvered or something else unexpected happens, so that our orbit model does not describe the true orbit accurately enough. This is especially needed for geostationary satellites of the Beidou system, as we will explain in Section 6.5.

When we have only one broadcast available, i.e. the first received or the first one after reset, we do not use latent forces yet in the prediction. This is because we have not had enough data and estimation is not yet accurate enough. Once the second broadcast has been received, and the data has been used for estimating the latent forces with Extended fixed-point smoother, we start using latent forces in predictions also.

We started from GPS week 1887 and predicted 20 weeks forward. This corresponds to the first few months of 2016. This is then repeated 50 times. This gives us enough data to do reliable analysis on accuracy.

6.3 Results for GPS

First we take a look at one example of GPS satellite PRN 16. SISRE is calculated with both methods with respect to time and results are shown in Figure 6.1.

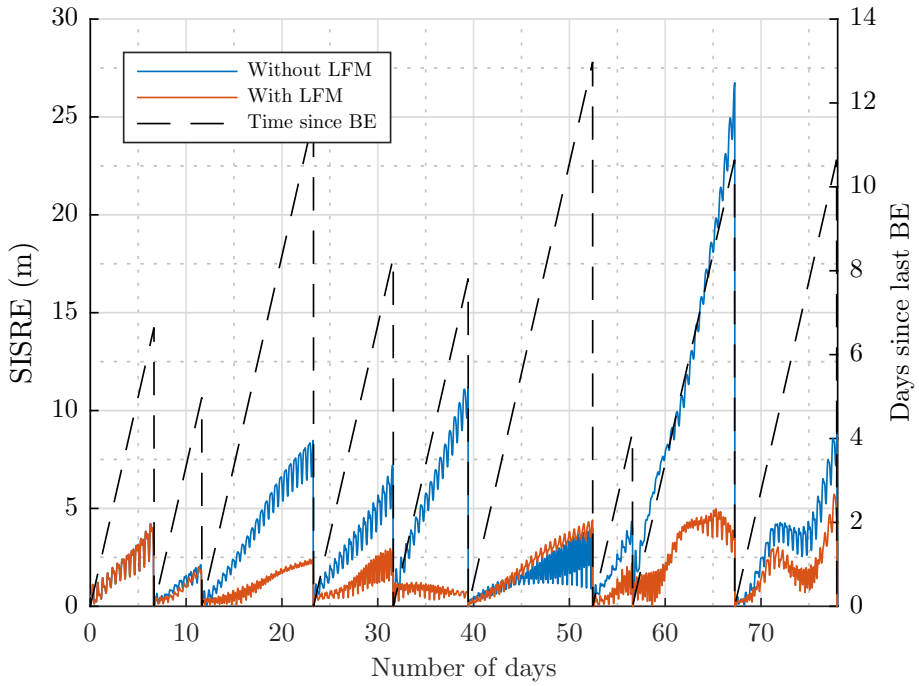

Figure 6.1 *Example of SISRE accuracy comparison with and without use of latent force model for GPS satellite PRN 16. Prediction started from GPS week 1879. Once a few broadcasts have been received, predictions get much more accurate when using LFM. The scale for y-axis for the blue and red curves is on the left while the one showing the age of the last received broadcast is on the right.*

We see that once a few broadcasts have been received, latent force model decreases the error vastly. This particular example is an exceptionally good case, but it

demonstrates that latent forces can be estimated from broadcast ephemeris data and this can be used to improve the prediction accuracy.

For GPS satellites LFM improves the predictions for all satellites. Some satellites improve more than others but it is clear that all satellites benefited from using LFM. From Table B.3, we see that even the smallest improvement in 68% quantile at 7-day prediction is 12.6%, while the largest is astonishing 53.3%.

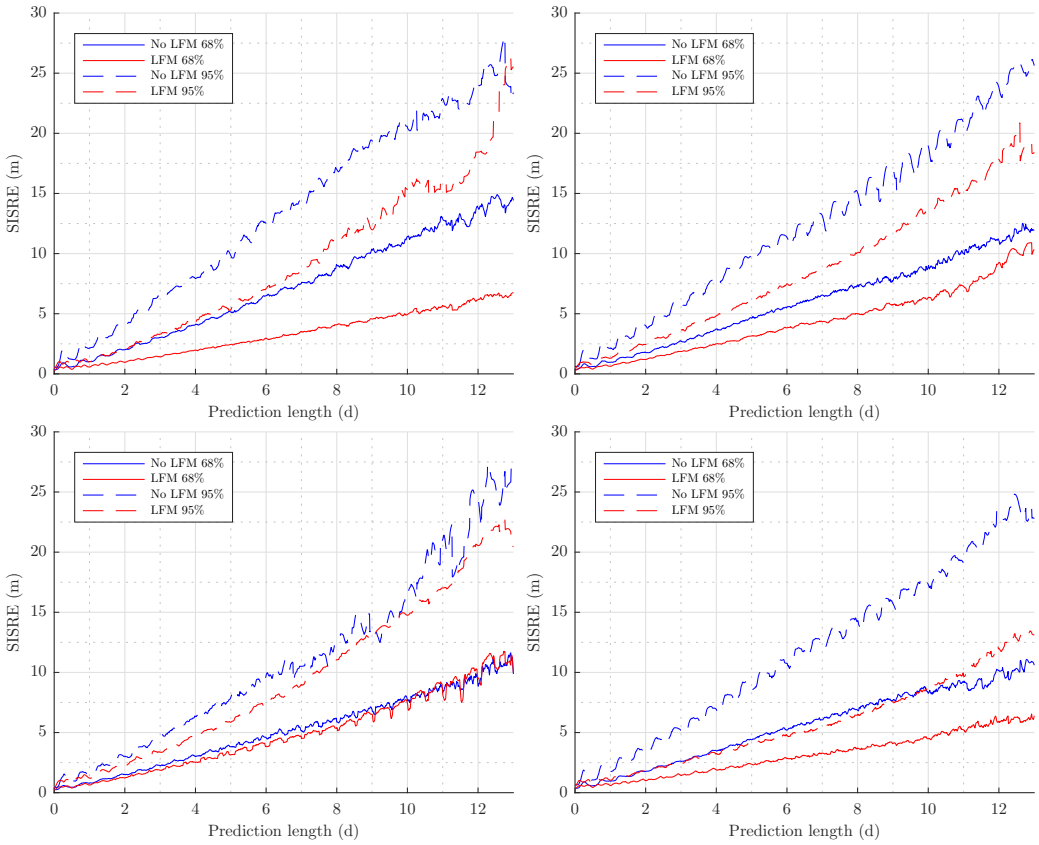

Figure 6.2 *A few examples of* 68% *and* 95% *quantile curves with and without LFM for GPS satellites. Prediction during the last day is cut off because beyond this point there are too few prediction values to give reliable error statistics. Satellite PRNs are: top-left: 16, top-right: 1, bottom-left: 5 and bottom-right: 25.*

As maximum interval, we used two weeks. This also means that maximum prediction length is two weeks. Results from the figures are reliable to at least around 10–12 days, since as we mentioned earlier, as we get closer to the maximum interval length, we have fewer predictions.

Different satellites of course behave differently and not every satellite is as good as this. We have 68% and 95% quantile comparison of the standard approach and LFM for a few satellites in Figure 6.2.

We see that while LFM does not affect all satellites as much, it usually improves predictions significantly. This can also be seen in Table B.3. For most satellites, the improvements in percentages are around $20 - 30\%$, and for some satellites even over 50%. This is a far bigger improvement than the few percentage units achieved by adding analytical expressions to the force model [28, 27].

6.4 Results for GLONASS

With GLONASS initial state estimation is normally done a bit differently than with GPS. Because broadcast ephemerides have such short validity interval, two broadcasts are usually used to estimate the initial state. We shall compare the performance of LFM to this approach rather than using only one BE. The difference between the two broadcasts is always 12 hours.

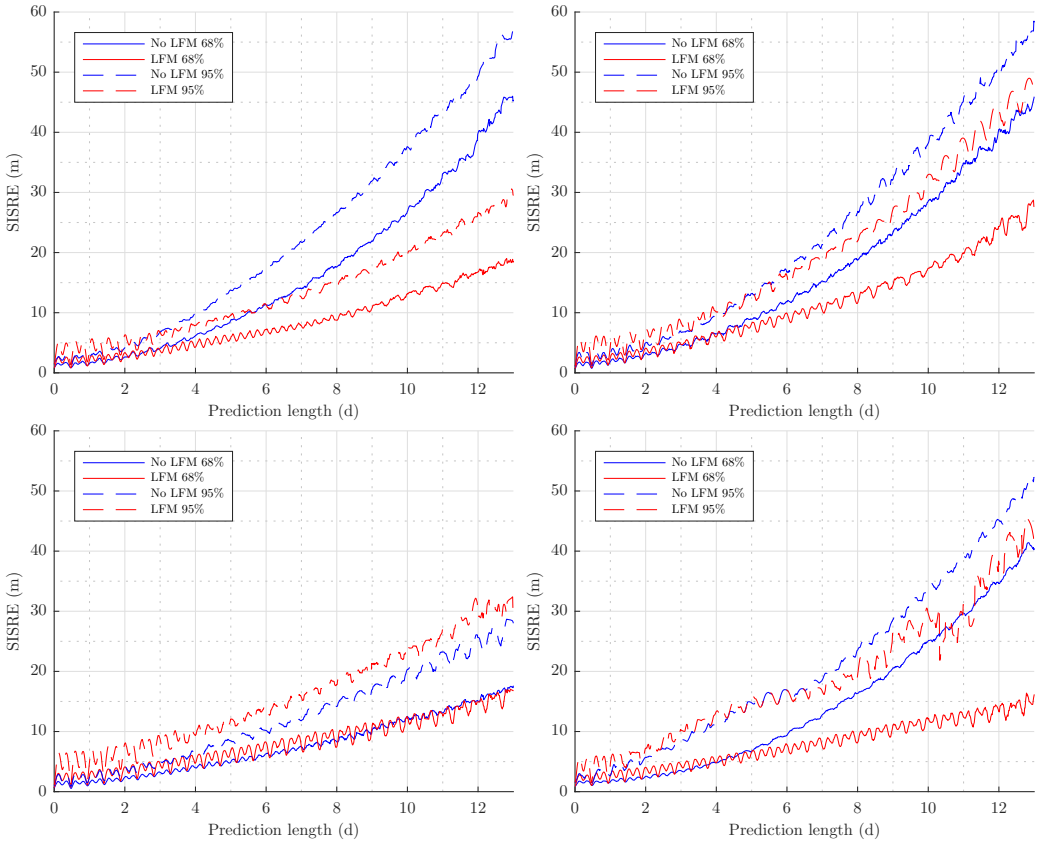

Figure 6.3 *A few examples of* 68% *and* 95% *quantile curves with and without LFM for GLONASS satellites. Performance with GLONASS is very satellite-dependent. Satellite PRNs are: top-left: 19, top-right: 14, bottom-left: 20 and bottom-right: 24.*

With GLONASS, LFM showed the least improvement. In some cases, the predictions became slightly worse, in some cases predictions improved.

In Figure 6.3 we show some results for GLONASS satellites. We have collected different looking quantile curves to demonstrate the satellite-dependent behaviour of LFM with GLONASS.

When comparing for example satellites PRN 19 and PRN 20, we see that LFM works well with the former whereas with the latter predictions become slightly worse. One reason for this might be that we are already using two broadcasts to estimate the initial state. Therefore we do not get the same advantage of multiple broadcasts that we get with other constellations.

Again, full results for GLONASS can be viewed from Table B.2.

6.5 Results for Beidou

Beidou has three different satellite types: MEO, IGSO and GEO. The prediction accuracy has previously been the best with MEO, then IGSO and the worst with GEO satellites. This general behaviour of different satellite types can also be seen in following figures.

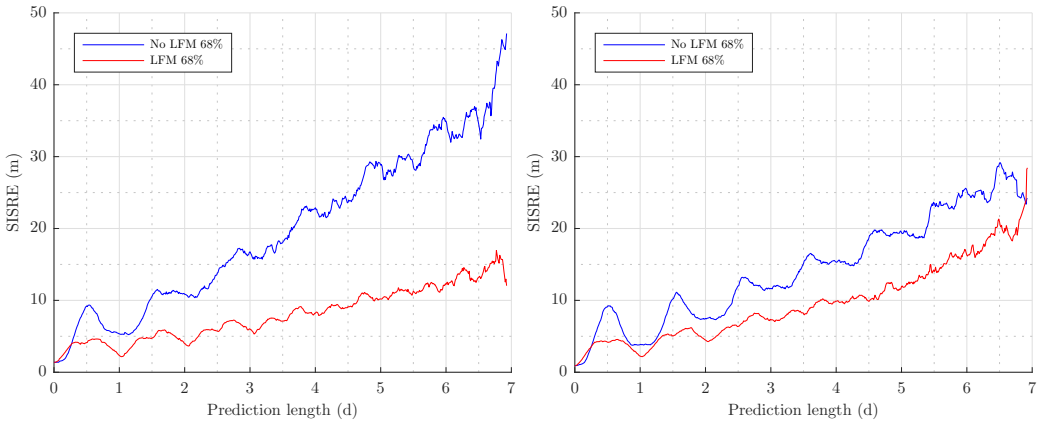

Figure 6.4 *Two examples of GEO satellites. Only* 68% *quantiles are shown. The left figure (PRN 4) shows a satellite for which LFM worked exceptionally well. In the right figure (PRN 5) we see a more typical behaviour for a GEO satellite.*

For Beidou GEO satellites it is noted that roughly every 25-35 days, an orbital maneuver is made for station-keeping purposes [34]. Predictions over these maneuvers are quite difficult to do reliably and often prediction fails. This also means that LFM is reset every time such event occurs. Since we need a few broadcasts for the filter to converge, LFM does not perform as well with GEO satellites. This can also be seen in quantile curves and this is why we restrict ourselves to 68% quantile in GEO case.

With a maximum interval of two weeks we do not really see improvements in predictions (except with satellite PRN 4, which was exceptionally good). With one week maximum interval, the improvement becomes more clear. Comparison of one-week predictions with and without LFM is in Figure 6.4.

For IGSO and MEO satellites, we do not have similar unpredictable behaviour and we may also plot 95% quantile curves. For IGSO and MEO satellites, we may also use a two-week maximum interval. Some satellites do not show much improvement but this did not change with shorter prediction periods. With IGSO, we show only one-week predictions, since quantile curves after that in many cases become unstable.

In Figure 6.5 we see examples of two IGSO satellites. Improvements are not as good as with GPS or MEO satellites but still the improvement compared to our previous approach is quite good.

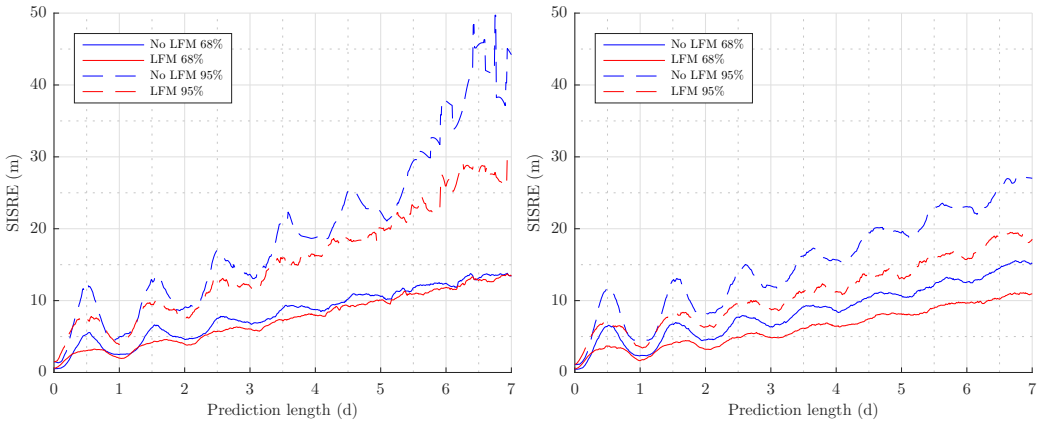

Figure 6.5 *Two examples of IGSO satellites. All IGSO satellites exhibited quite similar behaviour. Satellite PRNs are: left: 9 and right: 10.*

Improvement was biggest with MEO satellites. All tested satellites showed very large improvements. This can be seen for two satellites in Figure 6.6. Especially satellites PRN 11 and PRN 12 showed large improvements since 95% quantile with LFM was less than original 68% quantile. Also for these two satellites, the 68% quantile at two weeks was quite close to 10 meters which is very good compared to previous performance.

Detailed results are in Table B.1.

We conclude with a summary of performance comparison for each constellation. In Table 6.2 we show accuracy of the old and new methods as well as change in accuracy. For each constellation and for Beidou also for each satellite type we show 68% and 95% quantiles at some prediction length. We can see that general

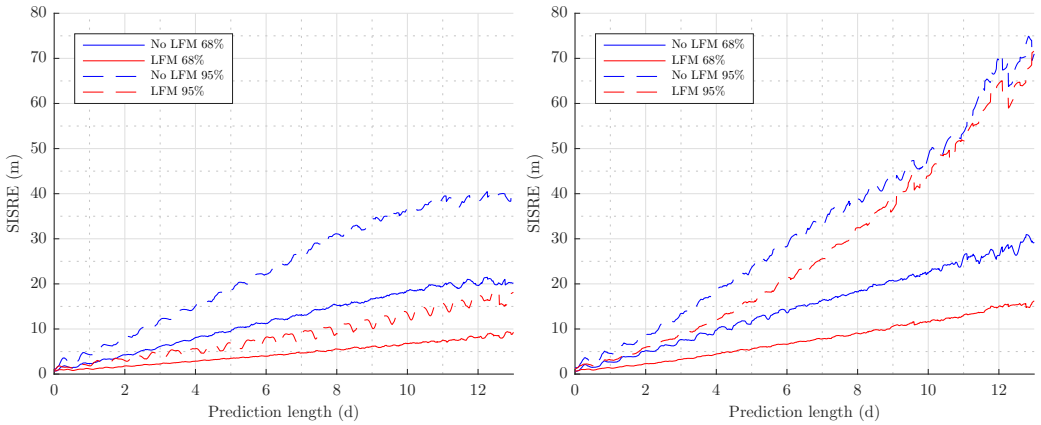

Figure 6.6 *Two examples of MEO satellites. Satellite PRNs are: left: 11 and right: 14.*

performance level is improved by some dozens of percentages depending on the constellation. In conclusion, LFM works with all different satellite types just as long as we make sure that frequencies of the oscillators correspond to the orbital period of the satellite.

68% quantile			95% quantile		
GPS/7 days					
old	new	change	old	new	change
6.88 m	4.33 m	-37.1%	17.66 m	9.73 m	-44.9%
GLONASS/7 days					
old	new	change	old	new	change
12.16 m	9.32 m	-23.4%	28.11 m	27.85 m	-0.95%
Beidou MEO/7 days					
old	new	change	old	new	change
13.09 m	5.44 m	-58.48%	26.35 m	14.53 m	-44.86%
Beidou IGSO/7 days					
old	new	change	old	new	change
15.09 m	12.16 m	-19.39%	32.85 m	25.81 m	-21.4%
Beidou GEO/3 days					
old	new	change	old	new	change
15.11 m	7.21 m	-52.3%	-	-	-

Table 6.2 *Results of LFM for all constellations*

6.6 Weaknesses in the model

Our model for the latent forces is relatively simple and hence, it does have some weaknesses. Here we assess possible errors and deficiencies of our data-driven force model with a few additional simulations.

First we will consider the computation time. The most time-consuming part of the process is propagating the state and the covariance matrix from time of one broadcast to another. This is the part of the computation we will consider more detailed here. The covariance matrix, which needs to be propagated from previous ephemeris to the next grows rapidly in size as we increase the number of components. Our integrator in this work was Runge-Kutta-Fehlberg 7(8). Detailed description of the integrator can be found in [23]. Number of elements in covariance matrix increases quadratically as a function of N. This causes the computation time to also grow roughly at this rate.

The problem can be partially solved by integrating only the non-zero elements of the covariance matrix. Using this method causes the computation time to grow, in theory, only linearly. We verify this with simulations. The absolute values for computation times presented here should not be viewed to be too significant as simulations were run on a parallel cluster on Matlab, and true running time of the algorithm may vary vastly depending on programming language, code optimization and hardware. Relative computation times can, however, tell us something about the growth rate of computation time as N increases.

In Table 6.3 we show computation time with different number of latent force components. In these simulations, we used the same N for all directions and median time of 200 one-week simulations was taken as average time. Simulations were run with 64-bit Matlab (R2016a) on MacBook Pro version 10.12.3 with 8 GB 166 MHz memory and 2.8 GHz Intel Core i7 processor. The growth in computation time is almost linear, except sometimes we see vast jumps in computation time. This happens for example when increasing the number of components from $N = 5$ to $N = 6$. When $N \leq 5$, the computation time is still reasonable. We chose $N = 3$ in this work.

The critical factor in the computation time was the maximum number of components in one direction. When we had $N_{\mathrm{R}} = N_{\mathrm{T}} = N_{\mathrm{N}} = 3$, the computation time was much smaller than with configuration $N_{\mathrm{R}} = 7, N_{\mathrm{T}} = N_{\mathrm{N}} = 1$. In both cases, we had the same total number of components and therefore our state vector and covariance matrix are equally sized. This implies that large number of harmonic components in one direction causes our integrator to perform much slower. Some stopping criteria in the integrator may be the reason for this.

Number of components (N)	Average computation time (s)
1	8.9
2	10.8
3	12.2
4	13.6
5	15.4
6	44.4
7	48.6

Table 6.3 *Computation times with different number of latent force components.*

Our model is dependent on the data received from the satellite. If there is something wrong with the data, this will directly affect our latent force model. This we noted already when considering Beidou GEO satellites, when we frequently had to reset our latent force filter. Also, some forces, which are not included in our analytical force model, are not suitably modelled by assumptions in our latent force model, and so they can affect the estimation process and cause errors in the estimation.

For simplicity, we consider only GPS satellites here, and take one example satellite to show typical behaviour. We consider a 10-week period, where broadcast ephemerides are given to the filter every two days and compare errors with and without latent force model. We start from GPS week 1890, and our example satellite is PRN 1.

In Figure 6.7 we show a comparison of radial, transverse, and normal error components with and without latent force model. From the figure, we can make a few interesting observations.

First of all, when we use LFM, radial and transverse errors seem to get smaller, while normal error gets larger. From the point of view of reducing SISRE, and hence improving positioning accuracy, this does not cause problems, since normal error contributions to SISRE are usually practically zero, and the difference is not large enough to make a difference. This is, however, an interesting feature from a theoretical standpoint. For some reason, superposition of resonators does not describe the missing forces in normal direction as accurately as if does with the other two directions. In this example, the other two decrease significantly, which is beneficial from viewpoint of positioning accuracy.

Another interesting feature is that error without LFM has clear cyclic behaviour. This is clearly visible from the transverse error component; errors tend to same direction for days or even weeks. This indicates that there are error sources, that do not have a frequency of one orbital revolution, or its harmonic component. Our latent force model, however assumes that there are only this sort of forces missing

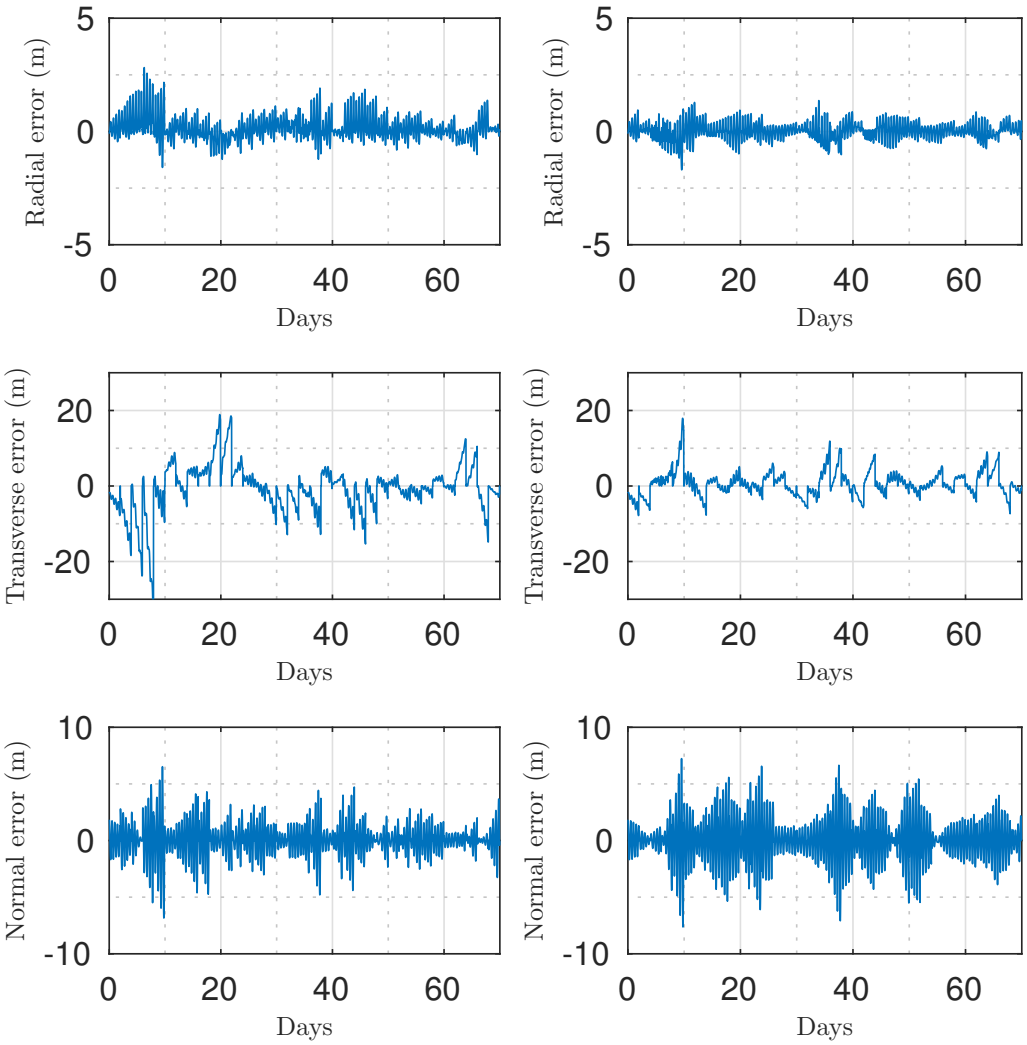

Figure 6.7 *Comparison of error components with and without LFM. On the left side we have prediction errors without any latent forces and on the right side we use our latent force model.*

and hence, it can model the missing forces erroneously.

We could, of course, include any periodic force into our latent force model in a similar manner as done in this work. We could identify some error sources, one of which could be the orbital period of the Moon, and include corresponding periodic components into our latent force model. Similarly, we could model oscillating force components to the direction of the Sun.

If we wanted to precisely model all the missing forces, we would most likely have to partially abandon our periodicity assumption, or at least include some other model for the missing forces. In this case, we would have to model the missing forces with

some Gaussian process assumption, and study how the forces behave at different times and in different directions. Based on this, an alternative model could be created for them. We, however, omit this sort of analysis in this study and leave it as a topic for future research.

7. CONCLUSIONS

The purpose of this research was to study if satellite orbit predictions can be improved by introducing additional data-driven terms and estimate them with the data received directly from a satellite. We presented a model where additional terms to force model were modelled as sum of harmonic oscillation components that correspond to the orbital period of the satellite, and concluded that this relatively simple model can be used to improve orbit predictions. In addition, different constellations do not need different models but the same model works for all, as long as we adjust the base frequency to match the orbital period of the satellite. This method, however, requires storing ephemeris data in the positioning device, since one broadcast is not enough to estimate the latent forces.

The latent force model used in this work was rather simple. We noted a few deficiencies that our model has and also considered how a more accurate model could be constructed. More complicated models could improve the prediction accuracy even further. Downside is that computation time might increase too rapidly. This was already one of the reasons, why the number of components in each direction was chosen to be small. Adding more components or increasing the complexity of the model some other way would most likely increase computation even further. For applications and positioning devices, where this is not a problem, such approach could be considered.

We considered the estimation of additional forces with broadcast data. Naturally, if we used precise data, we could get more accurate results. Since we already assume that some parameters are provided to positioning device as assistance data, it would be reasonable to consider, could the information about latent forces also be delivered as this sort of data. This way, we could have a more accurate estimate of the forces, since we could increase the number of components and estimate the forces based on lots of precise data. The problem with this approach is that number of parameters to be delivered to the device would probably be too many, since we need both the estimate and the covariance matrix of the state. If this information could be compressed into small number of bits, or the data size is not a restriction in delivering assistance data, this is certainly an approach worth considering. This way computationally heavy algorithms wouldn't have to be done in the device.

In this work, we considered only the accuracy of the orbit. In reality, the accuracy of the positioning depends also on the accuracy of the clock. If the clock accuracy is much worse than the corresponding orbit accuracy, the improvements in orbit prediction do not help that much. Naturally, if clocks on the other hand are stable and easily predictable, then improvements in orbit accuracy also result in improvements in positioning accuracy. In our research, we have multiple methods for predicting the clock offset, but generally the orbit and clock offset errors are on the same level, which means that improving either one improves the total prediction accuracy. If the true improvement of the method is to be considered, the clocks should also be taken into account. These results, however, provide quite good theoretical standpoint for the orbit improvements and clock stabilities can be easily tested separately.

It should also be noted that the forces that are estimated from broadcast data should always be tested to be working for a prediction implementation. The forces after all are meant to compensate the deficiencies of the model. If the force model is vastly different to the one presented in this work, it is possible that the model for latent forces presented in this work does not improve the predictions.

We did not aim to create a perfect model for the missing forces, but rather create a general model that improves the prediction accuracy. As future work, we left for example finding a more complex model for the latent forces. The missing forces could be studied as Gaussian processes with certain priors to get more information about their nature. We studied only oscillators with harmonic frequency corresponding to one orbital period of a satellite. Longer periods such as the period of Moon's orbital period around the Earth or Earth's orbital period around the Sun could be studied. We also used only broadcast ephemerides in constructing the model. If precise ephemerides were available, a more complex model could be used.

We can conclude that if the orbit predictions are based mostly on broadcast data, that once the few major forces have been taken into account, the biggest improvement can be achieved by adding empirical force models that are adjusted with the broadcast data, assuming that data is somewhat regularly received, and device sometimes has access to assistance data delivered from the server. This gives a better improvement than the traditional approach of adding more terms to the analytically formed force model.

BIBLIOGRAPHY

[1] "IERS earth orientation data," [Accessed: 6.2.2017]. [Online]. Available: https://www.iers.org/IERS/EN/DataProducts/EarthOrientationData/eop.html

[2] "MGEX products," [Accessed: 8.2.2017]. [Online]. Available: http://mgex.igs.org/IGS_MGEX_Products.html

[3] "Positim high accuracy GNSS solutions and services," [Accessed: 5.1.2017]. [Online]. Available: http://www.positim.com/navsys_overview.html

[4] "Global navigation satellite system GLONASS interface control document," Coordination Scientific Information Center, Tech. Rep., 1998.

[5] *Global Positioning System Standard Positioning Service Performance Standard*, 4th ed., September 2008.

[6] "Global Positioning Systems Directorate Systems Engineering & Integration Interface Specification IS-GPS-200," Tech. Rep., December 2015.

[7] "RINEX - the receiver independent exchange format, version 3.03," International GNSS Service (IGS), RINEX Working Group and Radio Technical Commission for Maritime Services Special Committee 104 (RTCM-SC104), Tech. Rep., July 2015.

[8] J. Ala-Luhtala, M. Seppänen, S. Ali-Löytty, R. Piché, and H. Nurminen, "Estimation of initial state and model parameters for autonomous GNSS orbit prediction," in *International Global Navigation Satellite Systems Society Symposium 2013 (IGNSS2013)*, Gold Coast, Queensland, Australia, July 2013.

[9] J. Ala-Luhtala, M. Seppänen, and R. Piché, "An empirical solar radiation pressure model for autonomous GNSS orbit prediction," in *Proceedings of PLANS 2012 IEEE/ION Position Location and Navigation Symposium*, April 2012, pp. 568–575, [Accessed: 12.1.2017]. [Online]. Available: http://URN.fi/URN:NBN:fi:tty-201311011414

[10] M. Alvarez, D. Luengo, and N. D. Lawrence, "Latent force models," in *Proceedings of 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[11] M. Alvarez, J. R. Peters, N. D. Lawrence, and P. B. Schölkopf, "Switched latent force models for movement segmentation," in *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp.

55–63, [Accessed: 25.2.2017]. [Online]. Available: http://papers.nips.cc/paper/4001-switched-latent-force-models-for-movement-segmentation.pdf

[12] M. A. Alvarez, D. Luengo, and N. D. Lawrence, "Linear latent force models using Gaussian processes," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, November 2013, pp. 2693–2705.

[13] K. Burrage, I. Lenane, and G. Lythe, "Numerical methods for second-order stochastic differential equations," *SIAM J. on Scientific Computing*, vol. 29, no. 1, pp. 245–264, 2007.

[14] L. Chen, W. Jiao, X. Huang, C. Geng, L. Ai, L. Lu, and Z. Hu, *Study on Signal-In-Space Errors Calculation Method and Statistical Characterization of BeiDou Navigation Satellite System.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 423–434, [Accessed: 15.12.2016]. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37398-5_39

[15] S. P. Diaz, M. Meurer, M. Rippl, B. Belabbas, M. Joerger, and B. Pervan, "URA/SISA analysis for GPS-Galileo ARAIM integrity support message," *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, pp. 735–745, September 2015.

[16] R. Fisher, "Description of JPL solar system ephemeris," [Accessed: 2.8.2016]. [Online]. Available: http://www.cv.nrao.edu/~rfisher/Ephemerides/ephem_descr.html

[17] J. Hartikainen and S. Särkkä, "Sequential inference for latent force models," *CoRR*, 2012, [Accessed: 2.2.2017]. [Online]. Available: https://arxiv.org/pdf/1202.3730.pdf

[18] J. Hartikainen, M. Seppänen, and S. Särkkä, "State-space inference for non-linear latent force models with application to satellite orbit prediction," 2012, [Accessed: 3.3.2017]. [Online]. Available: http://icml.cc/2012/papers/477.pdf

[19] A. Jazwinski, *Stochastic Processes and Filtering Theory*, ser. Mathematics in Science and Engineering. Academic Press, Inc., 1970, vol. 64.

[20] E. D. Kaplan and C. J. Hegarty, *Understanding GPS: principles and applications*, 2nd ed. Artech House, 2006.

[21] P. Maybeck, *Stochastic models, estimation, and control*, ser. Mathematics in Science and Engineering. Academic Press, 1982, vol. 3.

[22] P. Misra and P. Enge, *Global positioning system: Signals, measurements, and performance*, 2nd ed. Ganga-Jamuna Press, 2006.

[23] O. Montenbruck and E. Gill, *Satellite Orbits: Models, Methods and Applications*, 3rd ed. Springer, 2005.

[24] O. Montenbruck, P. Steigenberger, and A. Hauschild, "Broadcast versus precise ephemerides: a multi-GNSS perspective," *GPS Solutions*, vol. 19, no. 2, pp. 321–333, 2015, [Accessed: 5.1.2017]. [Online]. Available: http://dx.doi.org/10.1007/s10291-014-0390-8

[25] NGA. EGM2008 model coeffcients. [Accessed: 6.7.2016]. [Online]. Available: http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/first_release.html

[26] M. Poutanen, *Satelliittipaikannus*. Tähtitieteellinen yhdistys URSA, 2016.

[27] A. Pukkila, "GNSS sateliitin radan ennustamisen virhetermien analysointi," Master's thesis, Tampere University of Technology, 2015, [Accessed: 17.2.2017]. [Online]. Available: http://URN.fi/URN:NBN:fi:tty-201505081267

[28] A. Pukkila, J. Ala-Luhtala, R. Piché, and S. Ali-Löytty, "GNSS orbit prediction with enhanced force model," in *2015 International Conference on Localization and GNSS (ICL-GNSS)*, June 2015, pp. 1–6.

[29] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013, [Accessed: 13.2.2017]. [Online]. Available: http://becs.aalto.fi/~ssarkka/pub/cup_book_online_20131111.pdf

[30] S. Särkkä, A. Solin, A. Nummenmaa, A. Vehtari, S. Vanni, and F.-H. Lin, "Dynamic retrospective filtering of physiological noise in BOLD fMRI: DRIFTER," *Neuroimage*, vol. 60, issue 2, pp. 1517–1527, 2012.

[31] M. Seppänen, "GPS-satelliitin radan ennustaminen," Master's thesis, Tampere University of Technology, 2010, [Accessed: 10.10.2016]. [Online]. Available: http://URN.fi/URN:NBN:fi:tty-2011051914680

[32] M. Seppänen, J. Ala-Luhtala, R. Piché, S. Martikainen, and S. Ali-Löytty, "Autonomous prediction of GPS and GLONASS satellite orbits," *NAVIGATION*, vol. 59, no. 2, pp. 119–134, 2012.

[33] P. R. Spofford, "The national geodetic survey standard GPS format SP3," [Accessed: 15.2.2017]. [Online]. Available: ftp://igs.org/pub/data/format/sp3_docu.txt

[34] P. Steigenberger, U. Hugentobler, A. Hauschild, and O. Montenbruck, "Orbit and clock analysis of Compass GEO and IGSO satellites," *Journal of Geodesy*, vol. 87, no. 6, pp. 515–525, June 2013.

[35] M. K. Titsias and N. D. Lawrence, "Bayesian Gaussian process latent variable model," in *Proceedings of 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[36] X. Zhang and R. Piché, "Application of the Hill-Clohessy-Wiltshire equation in GNSS orbit prediction," in *2014 International Conference on Localization and GNSS (ICL-GNSS)*, June 2014, pp. 1–6.

A. MATRICES AND VECTORS USED IN LATENT FORCE MODEL

Model for latent forces:

$$\begin{aligned}\frac{d}{dt}\mathbf{c}_{\mathrm{X}}(t) &= \mathrm{F}_{\mathrm{X}}\mathbf{c}_{\mathrm{X}}(t) + \mathrm{L}_{\mathrm{X}}\mathbf{w}_{\mathrm{X}}(t)\\ u_{\mathrm{X}}(t) &= \mathrm{H}_{\mathrm{X}}\mathbf{c}_{\mathrm{X}}(t) + b_{\mathrm{X}} + \epsilon_{\mathrm{X}}(t)\end{aligned}$$

Augmented state-space model for the satellite:

$$\frac{d}{dt}\mathbf{x}_a(t) = \mathbf{f}_a(\mathbf{x}_a(t), t) + \mathrm{L}_a\mathbf{w}_a(t)$$

State in latent forces ($\mathbf{c}$, $\mathbf{c}_{\mathrm{X}}$, $\mathbf{c}_{\mathrm{R}}$, $\mathbf{c}_{\mathrm{T}}$, $\mathbf{c}_{\mathrm{N}}$, size $2N \times 1$):

$$\mathbf{c}(t) = \begin{bmatrix} c_1(t) & \frac{dc_1(t)}{dt} & c_2(t) & \frac{dc_2(t)}{dt} & \cdots & c_N(t) & \frac{dc_N(t)}{dt} \end{bmatrix}^T$$

State transition matrix in the latent force model (F, F_{X}, F_{R}, F_{T}, F_{N}, size $2N \times 2N$):

$$\mathrm{F} = \begin{bmatrix} 0 & 2\pi f & 0 & 0 & & 0 & 0 \\ -2\pi f & 0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & 4\pi f & & 0 & 0 \\ 0 & 0 & -4\pi f & 0 & & 0 & 0 \\ & & \vdots & & \ddots & \vdots & \\ 0 & 0 & 0 & 0 & \ldots & 0 & 2N\pi f \\ 0 & 0 & 0 & 0 & & -2N\pi f & 0 \end{bmatrix}$$

Dispersion matrix in latent force model (L, L_{X}, L_{R}, L_{T}, L_{N}, size $2N \times N$)

$$\mathrm{L} = \begin{bmatrix} 0 & 0 & & 0 \\ 1 & 0 & \dots & 0 \\ 0 & 0 & & 0 \\ 0 & 1 & & 0 \\ & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \\ 0 & 0 & & 1 \end{bmatrix}$$

Noise vector in latent force model ($\mathbf{w}$, $\mathbf{w}_{\mathrm{X}}$, $\mathbf{w}_{\mathrm{R}}$, $\mathbf{w}_{\mathrm{T}}$, $\mathbf{w}_{\mathrm{N}}$, size $N \times 1$):

$$\mathbf{w}(t) = \begin{bmatrix} w_1(t) & w_2(t) & \cdots & w_N(t) \end{bmatrix}^T$$

Measurement matrix in the latent force model (H, H_{X}, H_{R}, H_{T}, H_{N}, size $1 \times 2N$):

$$\mathrm{H} = \begin{bmatrix} 1 & 0 & 1 & 0 \cdots 1 & 0 \end{bmatrix}^T$$

Augmented state of the satellite ($\mathbf{x}_a$, size $(6 + 2(N_{\mathrm{R}} + N_{\mathrm{T}} + N_{\mathrm{N}}) + 3) \times 1$):

$$\mathbf{x}_a(t) = \begin{bmatrix} \mathbf{r}(t) \\ \mathbf{v}(t) \\ \mathbf{c}_{\mathrm{R}}(t) \\ \mathbf{c}_{\mathrm{T}}(t) \\ \mathbf{c}_{\mathrm{N}}(t) \\ b_{\mathrm{R}} \\ b_{\mathrm{T}} \\ b_{\mathrm{N}} \end{bmatrix}$$

State-transition function of the augmented state-space model ($\mathbf{f}_a$, size $(6 + 2(N_{\mathrm{R}} + N_{\mathrm{T}} + N_{\mathrm{N}}) + 3) \times 1$):

$$\mathbf{f}_a(\mathbf{x}_a(t), t) = \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{a}(\mathbf{r}(t), t) + \mathbf{a}_{\mathrm{latent}}(\mathbf{r}(t), \mathbf{v}(t), t) \\ \mathrm{F}_{\mathrm{R}}\mathbf{c}_{\mathrm{R}}(t) \\ \mathrm{F}_{\mathrm{T}}\mathbf{c}_{\mathrm{T}}(t) \\ \mathrm{F}_{\mathrm{N}}\mathbf{c}_{\mathrm{N}}(t) \\ \mathbf{0}_{3\times 1} \end{bmatrix},$$

where $\mathbf{a}(\mathbf{r}(t), t)$ includes forces presented in Chapter 3 and $\mathbf{a}_{\text{latent}}(\mathbf{r}(t), \mathbf{v}(t), t)$ is calculated according to (4.1).

Dispersion matrix of the augmented model (L_a, size $(6 + 2(N_\mathrm{R} + N_\mathrm{T} + N_\mathrm{N}) + 3) \times (3 + N_\mathrm{R} + N_\mathrm{T} + N_\mathrm{N} + 3)$):

$$\mathrm{L}_a = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times N_\mathrm{R}} & \mathbf{0}_{3\times N_\mathrm{T}} & \mathbf{0}_{3\times N_\mathrm{N}} & \mathbf{0}_{3\times 3} \\ \mathrm{R}_{\mathrm{RTN}}^{-1}(\mathbf{r}(t), \mathbf{v}(t)) & \mathbf{0}_{3\times N_\mathrm{R}} & \mathbf{0}_{3\times N_\mathrm{T}} & \mathbf{0}_{3\times N_\mathrm{N}} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{2N_\mathrm{R}\times 3} & \mathrm{L}_\mathrm{R} & \mathbf{0}_{2N_\mathrm{R}\times N_\mathrm{T}} & \mathbf{0}_{2N_\mathrm{R}\times N_\mathrm{N}} & \mathbf{0}_{2N_\mathrm{R}\times 3} \\ \mathbf{0}_{2N_\mathrm{T}\times 3} & \mathbf{0}_{2N_\mathrm{T}\times N_\mathrm{R}} & \mathrm{L}_\mathrm{T} & \mathbf{0}_{2N_\mathrm{T}\times N_\mathrm{N}} & \mathbf{0}_{2N_\mathrm{T}\times 3} \\ \mathbf{0}_{2N_\mathrm{N}\times 3} & \mathbf{0}_{2N_\mathrm{N}\times N_\mathrm{R}} & \mathbf{0}_{2N_\mathrm{N}\times N_\mathrm{T}} & \mathrm{L}_\mathrm{N} & \mathbf{0}_{2N_\mathrm{N}\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times N_\mathrm{R}} & \mathbf{0}_{3\times N_\mathrm{T}} & \mathbf{0}_{3\times N_\mathrm{N}} & \mathrm{I}_{3\times 3} \end{bmatrix}$$

Augmented noise vector ($\mathbf{w}_a$, size $(3 + N_\mathrm{R} + N_\mathrm{T} + N_\mathrm{N} + 3) \times 1$)

$$\mathbf{w}_a(t) = \begin{bmatrix} \mathbf{w}_\mathbf{x}(t) \\ \mathbf{w}_\mathrm{R}(t) \\ \mathbf{w}_\mathrm{T}(t) \\ \mathbf{w}_\mathrm{N}(t) \\ \mathbf{w}_\mathrm{b} \end{bmatrix}$$

B. COLLECTED RESULTS FOR EACH CONSTELLATION

Some satellites are not in the list. This is either because they were changed during the testing period or shortly before, so that parameters could not be estimated for these or data was missing for these satellites. Results are calculated in meters in orbit-only SISRE. If quantile curves start behaving abnormally (e.g. error $> 10^4$ m), results are also omitted. Prediction length is chosen so that it is reliable taking into account the maximum interval between two consecutive broadcasts. Results are presented as 68% and 95% quantiles in meters and accuracy change in percents. All results are rounded to three significant digits.

Beidou						
GEO/3 days						
PRN	68% old	68% new	68% change	95% old	95% new	95% change
1	15.2	7.96	-47.7%	-	-	-
2	13.4	8.02	-40.1%	-	-	-
3	18.5	8.31	-55.0%	-	-	-
4	16.8	6.02	-64.1%	26.2	27.9	+6.47%
5	11.6	7.25	-37.6%	-	-	-
IGSO/7 days						
6	14.6	13.8	-5.36%	-	-	-
7	17.4	11.9	-31.4%	45.3	40.9	-9.79%
8	15.6	10.6	-32.2%	32.5	34.1	+4.87%
9	13.4	13.4	+0.0676%	44.2	30.6	-30.9%
10	15.2	11.0	-27.9%	27.0	18.5	-31.3%
MEO/7 days						
11	13.2	4.72	-64.3%	26.7	8.74	-67.3%
12	9.35	4.77	-48.9%	19.8	7.92	-60.0%
13	-	-	-	-	-	-
14	16.4	7.83	-52.2%	33.3	25.5	-23.3%

Table B.1 *Results of LFM for Beidou satellites*

GLONASS/7 days

PRN	68% old	68% new	68% change	95% old	95% new	95% change
1	13.7	10.8	-20.9%	23.1	19.7	-14.8%
2	11.6	9.91	-14.4%	23.4	18.9	-19.2%
3	14.8	10.6	-27.9%	21.4	20.1	-6.29%
4	9.06	8.76	-3.30%	17.8	18.1	+1.58%
5	11.8	10.3	-12.3%	18.7	25.8	+38.1%
6	14.0	11.1	-20.7%	24.2	21.7	-10.4%
7	12.9	10.1	-21.9%	19.6	17.0	-13.2%
8	20.9	14.9	-28.8%	29.3	28.2	-3.68%
9	-	-	-	-	-	-
10	12.5	8.19	-34.6%	36.2	39.9	+10.2%
11	-	-	-	-	-	-
12	-	-	-	-	-	-
13	8.99	8.10	-9.93%	16.6	16.6	±0%
14	15.3	10.4	-31.7%	21.4	18.1	-15.3%
15	11.5	8.52	-25.9%	22.6	17.7	-21.8%
16	-	-	-	-	-	-
17	-	-	-	-	-	-
18	5.77	6.71	+16.5%	9.81	10.8	+10.1%
19	14.0	7.79	-44.5%	21.7	12.5	-42.2%
20	7.37	7.45	+1.13%	11.8	15.6	+32.7%
21	7.74	7.45	-3.71%	13.3	19.9	+49.6%
22	6.88	7.12	+3.43%	11.9	13.9	+16.8%
23	6.57	7.13	+8.50%	12.0	12.4	+3.55%
24	12.8	7.54	-41.0%	18.6	16.7	-10.2%

Table B.2 *Results of LFM for GLONASS satellites*

GPS/7 days

PRN	68% old	68% new	68% change	95% old	95% new	95% change
1	6.56	4.34	-33.9%	13.2	8.61	-34.5%
2	6.70	4.48	-33.2%	14.5	9.57	-34.1%
3	6.10	4.93	-19.2%	11.9	9.56	-20.0%
4	-	-	-	-	-	-
5	5.13	4.48	-12.6%	10.5	9.02	-14.2%
6	5.35	4.00	-25.2%	12.2	6.70	-45.3%
7	6.62	3.78	-42.9%	13.9	7.86	-43.4%
8	-	-	-	-	-	-
9	6.69	5.07	-24.2%	12.7	10.1	-21.0%
10	-	-	-	-	-	-
11	7.39	4.78	-35.3%	14.6	8.92	-38.9%
12	6.83	4.36	-36.2%	24.4	15.7	-35.7%
13	6.69	4.72	-29.5%	11.8	9.59	-18.5%
14	7.09	5.61	-20.8%	19.8	17.0	-14.2%
15	6.69	5.04	-24.6%	-	-	-
16	7.53	3.51	-53.3%	14.6	8.40	-42.4%
17	5.77	3.11	-46.1%	10.8	5.40	-50.0%
18	6.80	5.43	-20.1%	12.0	11.3	-5.89%
19	8.25	4.58	-44.5%	-	-	-
20	6.28	5.18	-17.5 %	11.5	10.2	-11.1%
21	6.55	4.46	-31.9%	15.1	7.88	-47.9%
22	6.30	4.93	-21.7%	19.1	15.6	-18.2%
23	6.44	4.26	-33.8%	11.9	9.01	-24.0%
24	5.72	3.85	-32.7%	13.7	7.16	-47.8%
25	7.12	4.37	-38.7%	-	-	-
26	7.54	3.95	-47.7%	13.1	9.31	-29.1%
27	6.83	3.32	-51.4%	-	-	-
28	6.77	4.77	-29.6%	-	-	-
29	6.05	3.28	-45.7%	12.5	5.44	-56.4%
30	6.47	3.92	-39.4%	15.5	7.25	-53.4%
31	7.18	3.74	-47.9%	15.1	9.03	-40.3%
32	-	-	-	-	-	-

Table B.3 *Results of LFM for GPS satellites*