UNIVERSITY OF CALIFORNIA,
IRVINE

Towards Opportunistic Navigation with LEO Satellites: Machine Learning for Satellite Orbit Propagation

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical and Computer Engineering

by

Jamil A. Haidar-Ahmad

Thesis Committee:
Professor A. Lee Swindlehurst, Chair
Professor Fadi Kurdahi
Professor Andrei M. Shkel

2022

DEDICATION

To my parents, Ali and Hanaa, and my sisters, Ghida and Elissa.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Zak Kassas, for all his time, guidance, and support over the past two years. I am especially grateful for all our discussions pertaining to research and life goals. Under his advisorship, I was able to develop as a researcher and mature as an individual.

I would like to thank my Master's committee members, Prof. A. Lee Swindlehurst, Prof. Fadi Kurdahi, and Prof. Andrei M. Shkel, for taking time out of their busy schedules to serve in the committee.

I would like to thank the Office of Naval Research (ONR), the Air Force Office of Scientific Research (AFOSR), the National Science Foundation (NSF), and the U.S. Department of Transportation (DOT) for supporting my research.

I would like to thank my Coding Night colleagues, Beverly, Floranne, Robert, Garrett, and Zongchang from the Graduate Electrical Engineering and Computer Science (GEECS) group, for allowing me to practice my coding skills and providing me with intellectually challenging yet entertaining problems and discussions.

I would like to thank my friends at the ASPIN Lab; Ralph Sabbagh and Sharbel Kozhaya for being great close friends who let me feel like I'm at home and who always introduced me to new and interesting science; Nadim Kairallah, for his mentorship, great companionship, and his support during long overnights; Alex Nguyen and Mu Jia for our very interesting conversations related to research or life in general; Mohammad Neinavaie for our times discussing his crazy yet genius ideas; and Mohammad Orabi, Ali Abdallah, and Dr. Joe Khalife for their amazing support and friendship which helped me go through tough times.

I would like to thank my aunts, Wafaa and Ghada for their unwavering support no matter the distance. I would also like to express my appreciation to my friends, the Orta, and family members who helped to shape me into the person I am today. Especially, I would like to give a very heartfelt thanks to my girlfriend, Joelle Oneissi, for her encouragement to do my best, unwavering support during difficult times, and helping me to become a better person.

Finally, I would like to thank my parents, Ali Haidar-Ahmad and Hanaa Awdeh, for their unconditional love and support throughout my life. No matter how busy they were, they always found time in the day to be there for me and always motivate me to become better academically and to grow as a person. Their wholehearted support is what motivates me to do my best every day. I am beyond grateful to have them both in my life, and none of my success would have been possible without both of them.

Finally, I would like to thank all the persons I encountered in my life who had an impact on me and shaped me into the person I am today.

ABSTRACT OF THE THESIS

Towards Opportunistic Navigation with LEO Satellites:
Machine Learning for Satellite Orbit Propagation

By

Jamil A. Haidar-Ahmad

Master of Science in Electrical and Computer Engineering

University of California, Irvine, 2022

Professor A. Lee Swindlehurst, Chair

Low Earth orbit (LEO) satellites inherently possess desirable attributes for navigation: (i) abundance, (ii) geometric and spectral diversity, and (iii) high received powers. However, the first prerequisite to satellite navigation is to know the satellites' ephemeris (i.e., position and velocity over time) and clock error states. Unlike global navigation satellite systems (GNSS), specifically designed for navigation, with satellites in medium Earth orbit (MEO) that constantly transmit ephemeris and clock corrections to users in their signals, LEO satellites, mainly operated by private companies, generally do not openly send such information in their proprietary signals. The quality of oscillators on-board LEO satellites' as well as their clock error states are completely unknown. Moreover, the most accurate publicly available information on LEO satellites' ephemerides is in the form of two-line element (TLE) files, which yield ephemerides with errors of a few kilometers in position and a few meters per second in velocity. Consequently, LEO satellites' states are completely unknown (clock errors) or uncertain at best (ephemeris).

This thesis addresses the aforementioned challenges by performing the opportunistic estimation of LEO satellites' states. First, a study of the use of machine learning for satellite orbital determination is conducted. Multiple models for orbit propagation are analyzed, and the model with the best performance is found. The model is then utilized in a STAN framework, experimentally demonstrating its capability of producing satellite ephemeris good enough to allow for desirable navigation solutions. Next, a framework is proposed to collect training data when the target ephemeris data is not available due to satellites not transmitting their ephemeris. Finally, the framework's feasibility is demonstrated experimentally first by localizing a stationary receiver and second by coupling an IMU with LEO observables to navigate a moving ground vehicle.

Chapter 1

Introduction

With the ongoing technological advances, constant connectivity and a shift to complete autonomy are already becoming necessary requirements when developing the technologies that shape our daily life. With these requirements comes the deeply ingrained reliance on the positioning, timing, and navigation (PNT) services that global navigation satellite systems (GNSS) have been providing for the past decades. Most critical national infrastructures depend heavily on GNSS availability, reliability, resilience, and integrity. This technology is utilized for industrial purposes as well as in our daily life, where it is applied in transportation (autonomous vehicles), urban air mobility, civil aviation, drone aviation and delivery systems, object and people tracking, agriculture, robotics, etc.

This overwhelming importance of the GNSS technology and our reliance on it raises cause for concern as any disruption could speak catastrophic throughout many infrastructures. Such disruptions are not purely theoretical concerns as there have been thousands of reports GPS jamming and spoofing as well as GNSS signals being unavailable (eg. under dense foliage, during unintentional interference, and in deep urban canyons). Furthermore, newer technologies require higher accuracy and resilience that GNSS cannot achieve in standalone mode, such as positioning and navigation indoors or in deep urban canyons. As a result, other technologies are necessary as alternative or complimentary to GNSS for guaranteeing the stringent PNT requirements of the current and future technologies.

Alternative signals that could be utilized for positioning and navigation purposes are called signals of opportunity (SOPs). Ambient SOPs are exploited and utilized in an opportunistic navigation paradigm that aims to overcome the limitations of standalone GNSS. These SOPs come in different forms and multiple sources ranging between terrestrial signals and extraterrestrial signals. Terrestrial SOPs include digital television, and AM/FM, cellular signals. Of particular note, the various generations of cellular signals (3G code-division multiple-access (CDMA), 4G long-term evolution (LTE), and 5G have shown the potential of meter-level level accuracy on ground and aerial vehicles [1, 2, 3, 4, 5, 6, 7].

Extraterrestrial signals, or space-based SOPs, are most commonly signals coming from Low Earth Orbit (LEO) satellites as they have the most desirable properties for use in a navigation framework [8, 9]. These LEO satellites have dominated the scene recently as more and more companies rush to enter the space race and launch their own mega constellations into space [10, 11]. This has been facilitated by the strides that have been achieved recently in technologies relating to building LEO satellites and launching them. These ground-breaking developments have dramatically decreased launch costs and have made available smaller and cheaper components to build satellites and other spacecraft. Furthermore, with the privatization of space launches, LEO satellites for mega-constellations can be launched from different places on Earth allowing the satellites to be placed directly into their designed orbits, instead of having to rely on a limited set of launch sites with limited time availability. Finally, these private companies have developed reusable orbital-class launch systems such as the Falcon 9 which dramatically reduced launch costs. With LEO space vehicles(SVs) already forming a virtual blanket covering Earth with around 3,800 active satellites in orbit, the space race is showing no signs of slowing down as these mega constellations have the potential to serve as the foundation for supporting new technologies and advancements in satellite imaging, remote sensing, and revolutionizing communication and navigation technologies such as 5G which demands higher data rates [12, 13, 14]. LEO satellites could also provide space-based optical mesh networks; global, high-availability, high-bandwidth, and low-latency internet [15, 16, 17]. Figure 1.1 shows some of the existing and future LEO megaconstellations that will form a virtual blanket cover around Earth. A direct consequence of LEO SVs' abundance is that their configuration relative to a receiver anywhere on Earth yields a low geometric dilution of precision (GDOP), which improves navigation accuracy. Figure 1.2 shows the current GDOP and a heatmap of the number of GNSS satellites over Earth. Compared to only one of the rising mega-constellations, Starlink, we can see a huge improvement in both GDOP and the number of visible satellites over Earth(availability) as illustrated in Figure 1.3c. Furthermore, LEO satellites are around twenty times closer to Earth than GNSS satellites, residing in medium Earth orbit (MEO), which results in the received power of

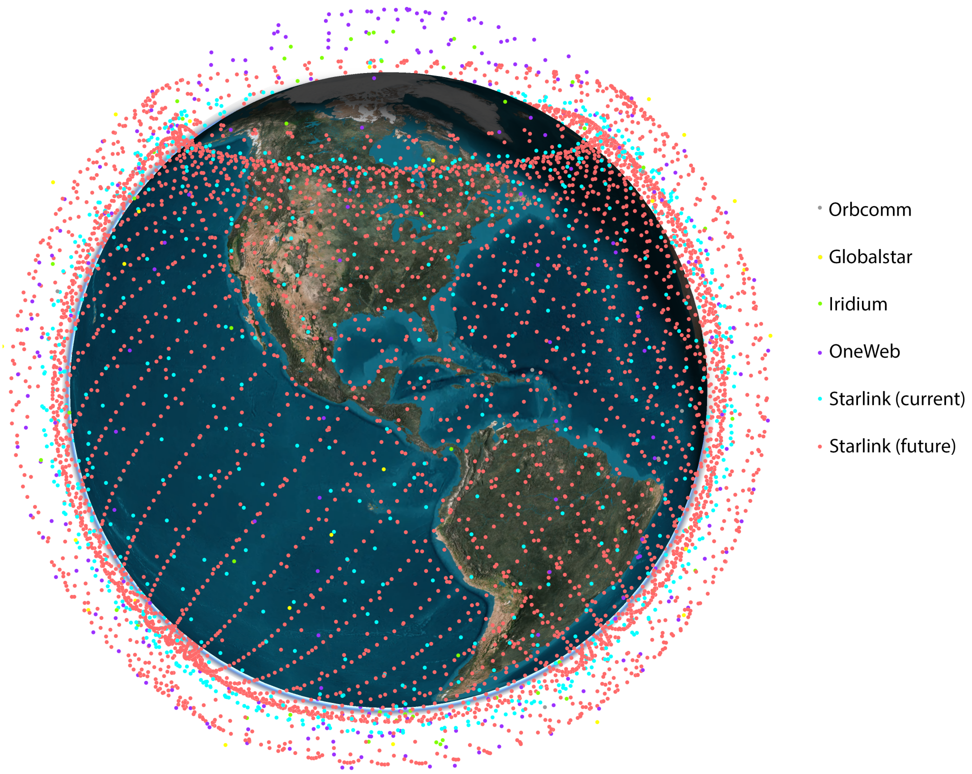


Figure 1.1: Existing and future LEO satellite constellations.

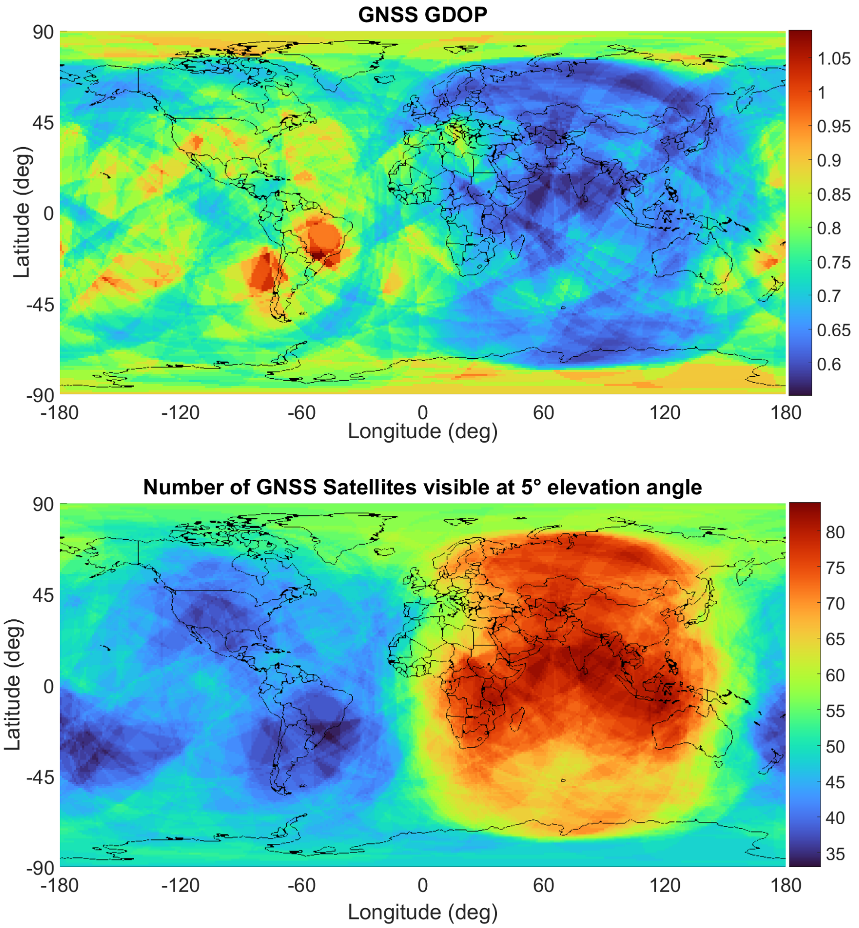
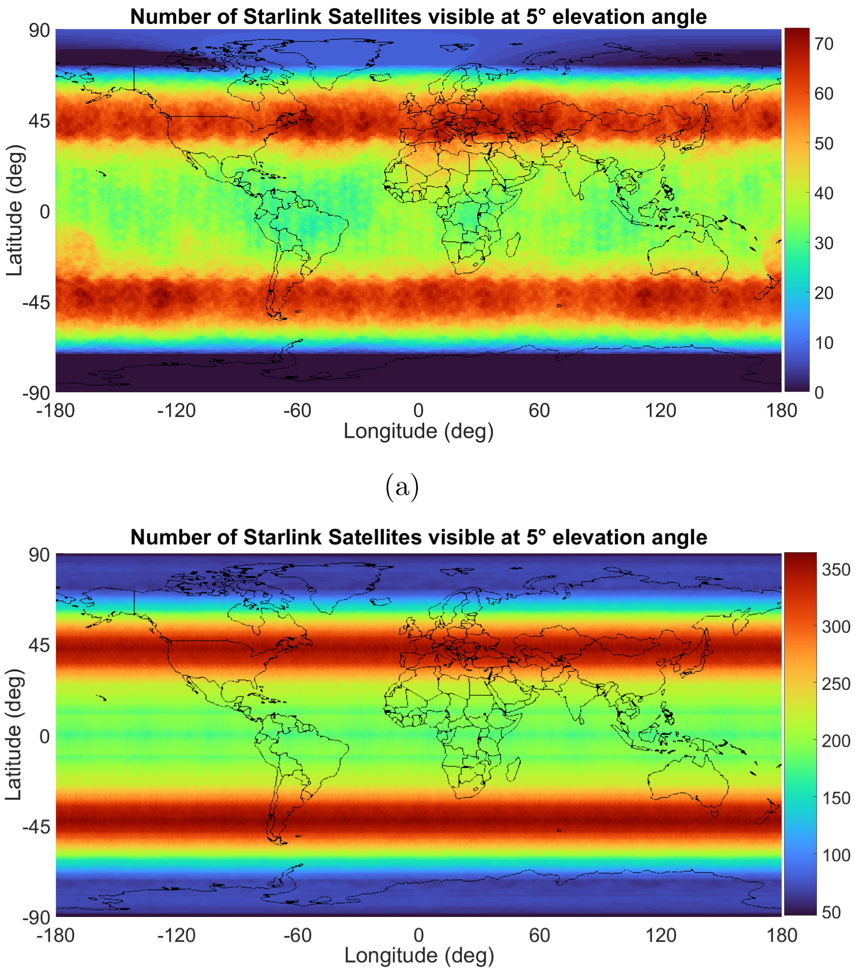


Figure 1.2: Current GNSS GDOP and number of available satellites heatmap.

LEO signals being up to 30dB more powerful than their GNSS counterparts [18, 19, 20]. Finally, LEO SVs transmit in a wide range of frequency ranging from 137–138 MHz in the very high frequency (VHF) band for Orbcomm to 11.325 GHz in the Ku band for Starlink; thus, improving the resilience of LEO SVs' signals to interference. LEO satellites' signals of opportunity could complement and safeguard GNSS to provide high levels of performance and operational resilience [21, 22, 23].

Some companies, such as Iridium in collaboration with Satelles, have taken advantage of that by launching satellites with highly accurate clocks and PNT (Positioning, Navigation, and Timing) capabilities. However, these services are not available for non-subscribers, and companies launching LEO satellites for communication, broadband Internet, and sensing



(a)

(b)

(c) Current (a) and future (b) Starlink number of available satellites.

purposes may not be willing to incur the expenses of integrating a PNT-capable system into their satellites and sharing the relevant information publicly [21, 22]. Fortunately, the LEO satellites' signals could be used opportunistically no matter if they were initially designed for navigation or not [24, 9, 25, 26]. LEO satellites' signals of opportunity (SOPs) could complement and safeguard GNSS with high levels of performance quality and operational resilience.

Utilizing LEO satellites signals opportunistically comes at the cost of: (i) developing specialized receivers to extract navigation observables from these signals, (ii) tackling clock errors of both satellite and receiver clocks which might not be as tightly synchronized as needed for navigation requirements, and (iii) knowing the satellite's ephemeris with minimal error. When it comes to the first point, the variability of signal structure and its unknown nature for each constellation makes it difficult to create receivers to extract navigation observables from these unknown signals. Furthermore, the equipment required must be able to capture these signals across multiple channels and frequencies, with specialized antennas, and must have the ability to sample signals at high sampling frequencies therefore handling data that must be kept synchronized while being processed at extremely high throughput in real time. This places a certain barrier-to-entry regarding equipment costs and the knowledge required for equipment selection. The first two challenges have been the subject of extensive research recently [24, 27, 9, 28, 29, 25, 30, 23, 31, 32, 33].

Several advancements were made to address the challenge of satellite orbital determination using models and algorithms for propagating the satellites' states as well as the associated uncertainty [34]. These propagators take into consideration, to various extents, multiple sources of perturbing forces such as the Earth's non-uniform gravitational field, atmospheric drag, solar radiation pressure (SRP), and third-body attraction (eg., the Sun and the Moon) [35]. The United States Air Force Simplified General Perturbations 4 (USAF SGP4) [36, 37] analytical propagator is based on a general perturbation analysis. SGP4 is used to generate ephemerides from a set of mean orbital elements given at a reference epoch in two-line elements (TLE) files, which are published and updated periodically by the North American Aerospace Defense Command (NORAD)[38]. However, these operational analytical orbit determination methods are based on limited dynamical models and mean elements which may not meet accuracy requirements for navigation [39, 40]. Space agencies usually employ high-precision orbit propagators (HPOP), which are numerical propagators used in conjunction with precise force models. However, numerical propagators require large amounts of data as well as significant computation time which renders them undesirable for real-time navigation purposes. Finally, semi-analytical models such as Draper semi-analytical satellite theory (DSST) combine the accuracy of numerical propagation and the characteristic speed of analytical propagation[41]. However, all of the previously mentioned propagators require sufficient prior knowledge of force model parameters as well as an accurate initial estimate, which is not readily available. The most accurate publicly available LEO satellite ephemeris information is in the form of TLE files that suffer from an error of a few kilometers, but with most error concentrated in the along-track axis of the LEO SV's body frame.

Machine learning (ML) has shown tremendous potential in radar and communications [42], and its powerful modeling capabilities have been recently studied to provide a less parameter-reliant orbit propagation solution [43, 44]. In [45, 46], distribution regression was used for orbital determination of objects in LEO and GEO space. Propagating LEO space debris orbits was studied through the use of support vector machines (SVMs) [47, 48], and LEO satellite orbital states were modeled using artificial neural networks (ANNs), SVMs, and gaussian processes (GPs). An exhaustive simulation study developed in [48, 49] showed that ANNs possess high regression capabilities compared to SVMs and GPs. Under the ANN approach exists some previous work that employed multiple neural network (NN) architectures, such as the Time Delayed Neural Network (TDNN) and Long Short-Term Memory (LSTM) neural networks [50]. However, utilizing ML in full orbital determination, allowing for completely replacing standard propagators, is yet to be achieved. Promising preliminary results were recorded in [51], in which a TDNN was trained using the data from two Orbcomm LEO satellites, which broadcast their three-dimensional (3-D) position in the Earth-centered, Earth-fixed (ECEF) coordinate frame from onboard GNSS receivers. However, utilizing ML in full orbital determination, allowing for completely replacing standard propagators, is yet to be achieved.

The contributions of this thesis are threefold and conduct a study on the use of machine learning in satellite orbital determination for use in a positioning and navigation framework.

First, the capability of computational neural networks and other machine learning algorithms in predicting satellite ephemeris time series (when given correct satellite ephemeris as training data) is studied:

- First, a comprehensive performance analysis is conducted between different ANN architectures for short and long-term satellite orbit prediction, such as TDNNs and LSTMs.
- Second, the performance of numerical and analytical propagators powered by a ML approach, to account for residual error correction, is studied.
- Third, the validation and generalization of the proposed ML-based orbital propagator for different orbits and constellations is assessed.
- Fourth, the performance of the proposed ML-based Simultaneous Tracking and Navigation (STAN) framework in estimating a ground vehicle's position is demonstrated experimentally.

Second, after ML is demonstrated to be able to predict ephemeris time series, the issue of not having correct satellite ephemeris to be used as training data is addressed. A hybrid numerical-machine learning approach is proposed for LEO satellite orbit prediction where the receiver has no prior knowledge on the satellite's position except for publicly available TLE files:

- A hybrid numerical-ML propagator is developed in a three-step framework: (i) refining a LEO satellite's ephemeris via opportunistic tracking initialized from an SGP4-propagated TLE, (ii) training the ML propagator on the refined ephemeris, thus extending the approach in [51, 52] to no prior true ephemeris information by solely relying on initialization from TLEs, and (iii) localizing a receiver opportunistically with the ML-propagated ephemeris.
- The ephemeris propagation performance of the hybrid numerical-ML framework is compared with that of standalone propagators using true decoded Orbcomm ephemeris data.

- Experimental results are presented to demonstrate the efficacy of the proposed framework with a first pass of an Orbcomm satellite during which the tracking to refine the ephemeris is performed and a second pass of the same satellite during which the opportunistic localization performance using ML-propagated ephemeris and SGP4-propagated ephemeris are compared.

Third, an experiment is conducted where a moving ground vehicle equipped with an inertial measurement unit (IMU) applies the proposed framework and utilizes LEO multi-constellation signals from up to 4 satellites at the same time to provide a navigation solution.

This thesis is organized as follows. Chapter 2 tackles the question of the capability of ML models in estimating satellite ephemeris. Chapter 3 presents experimental results for using ML-propagated ephemeris in a Simultaneous Tracking and Navigation (STAN) framework. Chapter 4 presents a comprehensive framework to first perform the tracking of LEO SVs, which are used to replace the unavailable true satellite ephemeris in training, and second to propagate that ephemeris and use it in localizing a stationary receiver. Chapter 5 presents an experiment conducted on a moving ground vehicle where the previously described framework is intergrated with an inertial measurement unit (IMU) to provide a tightly coupled IMU-navigation solution. The contributions of this thesis are summarized in Chapter 6.

Chapter 2

Machine Learning for Orbital Determination

2.1 Classical Orbital Determination

To address the challenge of satellite orbital determination, several orbital propagation methods were developed in the literature to propagate the states of satellites at any point in time [53, 54, 55, 56], and their differences can be categorized by two factors: complexity and accuracy. Numerous dynamics models which estimate the state of LEO satellites (position and velocity), as well as these estimates' uncertainty, have been developed over the years. The state of a satellite can be parametrized by its Keplerian elements, also known as Classical Orbital Elements (COE). Those orbital elements, along with some other information about a satellite' states, could be found in its corresponding Two-line Element sets (TLEs) which are publicly published on a daily basis by the North American Aerospace Defense Command (NORAD).

To this end, one way to mathematically propagate a satellite's ephemeris given its position and velocity vector in an inertial reference frame is to solve a second order differential equation relating the satellite's position and acceleration. This is also known as a Keplerian orbit, which is an unperturbed orbit where for any values of the initial state vector, the satellite's corresponding Keplerian orbit can be found.

However, propagating a satellite's ephemeris through unperturbed Keplerian orbital models numerically or analytically leads to highly inaccurate results as there are several sources of perturbing forces such as atmospheric drag, the Earth's oblateness causing a non-uniform gravitational field, solar radiation pressure, and other sources of gravitational forces (e.g., the Sun and the Moon). Therefore, the models realistically utilized are perturbed models, such as the two-body J2 propagator which takes into consideration the previously mentioned perturbations, including multiple forces which could offset the satellite's ephemeris from following an unperturbed path. These perturbations can be modeled as additive noise, which can still be handled through classical methods; however, these perturbations act as a cascade of nonlinear functions on the satellite's states.

Numerical methods such as the “Two Body” model, “Two-Body with J2” model [57], and the ”High-Precision Orbit Propagator” (HPOP)[58] are capable of producing highly accurate orbits; however, the J2 model is only accurate for short propagation periods, and HPOP requires a dense number of parameters for modeling perturbation forces as well as heavy computational loads to forecast the satellite orbit, rendering them unsuitable in real-time navigation systems. Meanwhile, their analytical counterparts, such as the Simplified General Perturbation (SGP4), are more computationally efficient and less parameter-heavy allowing for real-time propagation at the cost of introducing larger satellite position errors. The currently available parameters from NORAD, such as values from TLE files, do not provide enough data for achieving a desirable ephemeris accuracy, since utilizing these elements for orbital determination with a standard SGP4 propagator could lead to errors in the order of several kilometers, which accumulate and drift over time.

2.2 Machine Learning Overview

To address the issues of the unavailability of parameters for modeling perturbation forces and the nonlinearities introduced, a data-based approach is proposed. This is where the power of data-based ML methods can be utilized. ML attempts to build a model based on training data, which is subsequently utilized in several applications, such as making predictions or inferences, classifying data, or making decisions. ML has been utilized in solving highly dynamic, nonlinear, and highly uncertain problems. ML can be categorized into: (i) supervised learning, where the model is given both the input and expected output; (ii) unsupervised learning, where the model is only given the inputs; (iii) reinforced learning, where the model is given rewards or penalties according to the decisions it makes.

For orbital determination, a ML model would be given the task of taking data relevant to a LEO satellite's ephemeris as inputs and using this information to predict the satellite's future ephemeris; therefore utilizing supervised learning. Unlike deep learning models, where the feature extraction piece of the process is automated, enabling the use of large datasets, classical machine learning requires researchers to determine the set of features which would allow a model to understand the differences between structured data inputs. This work focuses on the classical approach as large amounts of reliable and highly accurate LEO satellite ephemerides data are not available. Furthermore, a large and complex deep learning model would take incrementally increasing time for both training and inference rendering it difficult to apply in real-time applications without specific hardware.

Predicting satellite ephemeris could be thought of as a time-series prediction problem. Utilizing classical methods for time series forecasting such as the Autoregressive Moving Average (ARMA) or the Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX) method would be straightforward approaches. These models assume a linear internal propagation model, which would be the case for a simple Keplerian orbit, such as the two-body model. Therefore, a better approach is to use neural networks which are capable of handling nonlinearities and even learning and storing hidden states that could encode parameters related to perturbation forces.

2.2.1 Feed Forward Neural Network

Artificial neural networks (ANNs), also termed "Universal Approximators", are theoretically capable of approximating any continuous function [59]. For time-series prediction, the ML model could be trained to find a function that takes as input previously known satellite ephemeris and tries to find a non-linear transformation to predict the satellite's ephemeris at the next time step. Consider some input $x \in \mathcal{X}$ that is mapped to $y \in \mathcal{Y}$ by an unknown function $f : \mathcal{X} \mapsto \mathcal{Y}$. Given enough compact subsets of data points mapping the input space and output space $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, the NN forms probability-weighted associations, effectively approximating the function that maps the two spaces. NNs are capable of achieving this by funneling the inputs through a multitude of "neuron" blocks, where every neuron computes $\sigma(x, w, b) = \sigma(w \cdot x + b)$, where w is the weight associated with every input, b is the bias associated with every input, and σ is the activation function of the neuron which could either be linear or could add nonlinearity to the system.

Once the estimated output $\hat{y}$ is computed, the error between the observed output y and $\hat{y}$ is calculated through a cost function $C(y, \hat{y})$, which the NN aims to minimize. There is an abundance of cost functions used in ML, each allowing for an intuitive understanding of the difference between the observed and estimated output. The mean-squared error (MSE) will be used as a cost function to assess the performance in the proposed approach, which can be expressed as

$$C(y, \hat{y})_{\text{MSE}} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad (2.1)$$

where n is the number of data points, y_i is the observed values, and $\hat{y}_i$ is the estimated values. Once the cost function is computed, each layer's weights and biases are updated according to how much those weights and biases contributed to the each layer's error output.

2.2.2 Recurrent Neural Network

While FFNNs create mappings from the input to output space, they lack the ability to retain memory from previous mappings. Instead of assuming that satellite ephemeris can be directly generated from previous inputs, a state space representation allows for the introduction of some sort of memory in the form of a hidden state. Recurrent neural networks (RNNs) model dynamic systems. A typical dynamic system is described by equations of the form:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

where x is the state, u is the input, and y is the observed output. In discrete time, we have:

$$x(t+1) = Ax(t) + Bu(t)$$
$$y(t+1) = Cx(t+1)$$

The goal here is system identification. An RNN model is capable of creating exactly this kind of structure in the discrete-time domain. The input is $u(t), u(t-1), \ldots u(t-N)$ for some prediction length N. The output is $x(t+1)$ at a single time. The point is that $x(t)$ is not memoryless, but in fact depends on the prior values of $x(t-n)$, memory of the last N samples, where this memory is stored in “state” x.

Each ”hidden unit” actually has memory of its own past state. So at time t, each hidden

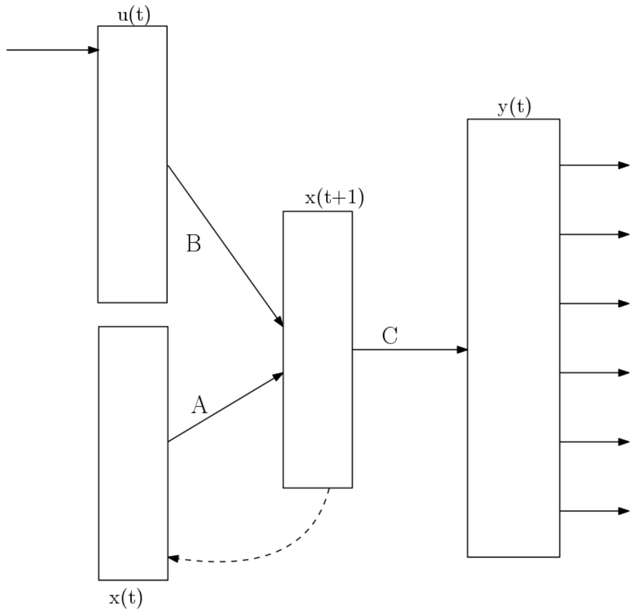


Figure 2.1: Simple RNN architecture

unit in the RNN looks at the current input $u(t)$ but also looks at its own prior value $x(t-1)$. There are weights from both inputs, so that $x(t) = Ax(t-1) + Bu(t-1)$. So this exactly implements the idea of a dynamic system. Finally, the output is read using another "dense" layer to obtain $\hat{y}(t) = Cx(t)$. Errors could be calculated at every time step comparing the true output to the predicted output $y - \hat{y}$, but typically this is only done for the "final" output, or after a finite number of time steps. This forces the network to learn a good update matrix A. Figure 2.1 shows the basic structure of a simple RNN.

Finally, "backpropagation through time"(BPTT) is used to figure out A and B. The hidden state $x(0)$ is initialized (usually zero, but could be taken from a previous estimate). Then that "forward propagates" through the A matrix a few times to get $x(t+N) = A^N x(0)$. At each step $Bu(t)$ is added in. The final error $y(t+N) - \hat{y}(t+N)$ is backpropagated through the A matrix (which is the same at each time step) and the B matrix back to different values of $u(t)$ at each time step, subject to the constraint that there is only one A and one B matrix. Figure 2.2 shows how an RNN is unfolded when the BPTT algorithm runs.

Note that this is different from the Kalman filter formulation, and closer to system identifi-

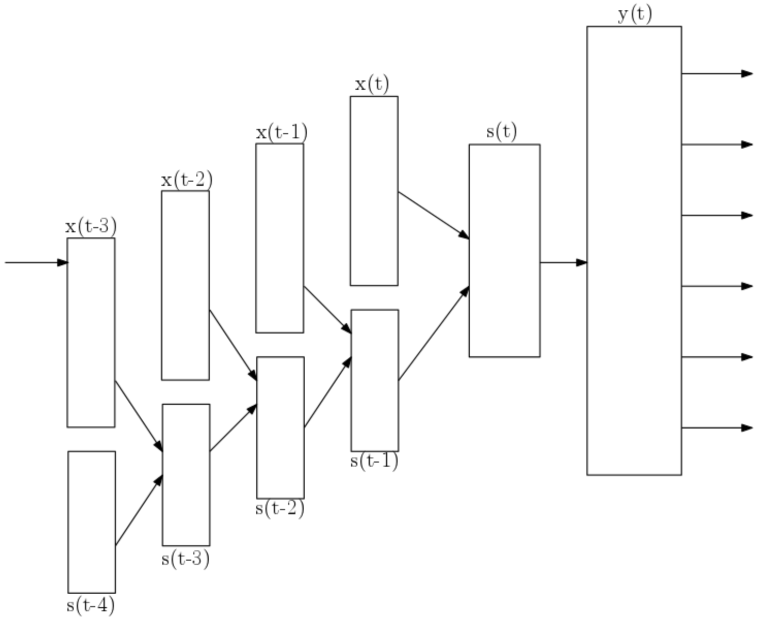


Figure 2.2: Unfolded RNN for BPTT.

cation, because the A and B matrices are learned, rather than assuming they are known in advance. At the same time the values for the hidden state x are learned.

2.3 Previous Work

Efforts towards less parameter-reliant models have been made in utilizing the powerful modeling capabilities of machine learning (ML) for providing an orbital propagation solution. In [45, 46], distribution regression was used for orbital determination of objects in LEO and GEO space. Propagating LEO space debris orbits was studied through the use of support vector machines (SVMs) [47, 48], and LEO satellite orbital states were modeled using ANNs, SVMs, and Gaussian processes (GPs). An exhaustive simulation study developed in [48, 49] showed that ANNs possess high regression capabilities compared to SVMs and GPs. Under the ANN approach exists some previous work that employed multiple neural network (NN) architectures, such as the Time Delayed Neural Network (TDNN) and Long Short-Term Memory (LSTM) neural networks [50]. However, utilizing ML in full orbital determination, allowing for completely replacing standard propagators, is yet to be achieved. Promising preliminary result were recorded in [51], in which a TDNN was trained using the data from two Orbcomm LEO satellites, which broadcast their three-dimensional (3-D) position in the Earth-centered, Earth-fixed (ECEF) coordinate frame from onboard GNSS receivers.

2.4 Proposed Frameworks

To achieve a computationally feasible ML solution for orbital determination and propagation, two frameworks which assume available training data are explored: (i) ephemeris prediction and (ii) error prediction. First, the ephemeris propagation framework presents an architecture that is capable of propagating a satellite's orbit from historical state data, where each state is given by $S = [r, \dot{r}]^\intercal$, and r and $\dot{r}$ are the LEO satellite's 3-D position and velocity in the ECEF frame. There has been previous work in training a TDNN to model a satellite's path for a short time window (approximately 30 seconds) in [51]. However, this model used the LEO satellite's states which are transmitted by onboard GPS receiver and transmitted in the downlink signal. While this approach produced promising results with Orbcomm LEO satellites, this cannot be generalized to other LEO satellite constellations, since they do not necessarily transmit their states openly. As such, a more realistic approach would be to train the NN to model a computationally heavy yet highly accurate propagator, such as HPOP. This model would look at historical HPOP data during training, and then take d previous consecutive states to output the next state: $\hat{S}_t = \Lambda(\hat{S}_{t-1}, \hat{S}_{t-2}, \ldots, \hat{S}_{t-d})$, where Λ is the function that maps previous states to the next state. A key constraint to the complexity of this model in allowing higher dimensionality is the time needed for inference, as the model should stay within real-time timing constraints.

Second, the error propagation framework models the error between a fast and less accurate propagator, such as SPG4, with a more accurate propagator, such as HPOP. This method is attractive since both propagators already handle the computation of highly dynamic parameters, leaving the NN to simply ”close the gap” between two propagators that already perform similarly. The model is trained to find the mapping from SPG4-propagated state vectors to HPOP-propagated state vectors $\hat{S}_{\text{HPOP}} = \Gamma(S_{\text{SGP4}})$, where Γ is the mapping between the two states.

While input data in the form of SGP4 ephemeris propagated from TLE files is readily available, target outputs (true satellite ephemeris) are not available for training. Therefore, a framework is developed to create these target outputs through satellite tracking and filtering.

2.4.1 Ephemeris Propagation

This section presents an ML model specialized for predicting LEO satellite orbits for short time windows.

Training

The proposed ANN models are investigated to study their ability and accuracy in predicting a LEO satellite's state vector (3-D position and velocity). Ground truth data in the ECEF reference frame was acquired using the Analytical Graphics System Tool Kit (AGI-STK) software, which is capable of generating highly accurate orbits through the use of the HPOP, a numerical propagator with high dimensionality and high fidelity [58]. AGI-STK allows for exporting such orbits for use in an ML environment. The satellite chosen for the simulation study is the Orbcomm F107 satellite (NORAD ID: 40087). AGI-STK provides its own database of satellite models, where this satellite's model parameters, such as its inertial mass distribution, are available. The toolkit provides an HPOP propagator with updated force models from its database. The force models include gravitational effects of the sun, moon, and options to include Jupiter, Venus, Saturn, and other planetary gravitational forces. The gravitational effects include Earth's gravitational model (2008) with high order and degree. Other forces include drag models, taking into account area/mass ratio, atmospheric density models, low altitude density models, and solar flux/geomagnetic models. Additionally, solar radiation pressure is utilized by the propagator while taking into account central body radiation pressure and eclipsing central bodies of the Earth and Moon. This makes the AGI-STK tool, utilizing HPOP, suitable for exporting propagated LEO satellite ephemeris to be used for training, validation, and testing [60].

Design

To achieve a ML model which could take initial historical data and propagating the satellite's orbit, using its previous output as new input, this model should be capable of time series prediction, as the satellite's state elements could be seen as its own time series. Feed forward neural networks (FFNN) [61, 62], which simply propagate from input to output in one direction. However, other NNs like recurrent NNs (RNNs), close the loop, retain memory, and provide a feedback to the NN. For this reason, the RNNs are expected to perform well in modeling highly nonlinear and harmonic data [63, 64, 65, 66].

TDNNs are simple FFNNs, except that in TDNNs the input is fed as a time series, which allows the NN to learn the dynamics of the system. For orbital determination, the output $\hat{y}(t)$ is the state vector $S_{\text{LEO}}(t)$ at time t, and the input is $I(0, 1, \ldots, N-1) = \{x(t), x(t-\tau), \ldots, x(t-(N-1)\tau)\}$ for N previous state vector data points. The TDNN's output is connected to the input I through a delay block, closing the loop, and effectively creating a nonlinear auto-regressive (NAR) prediction model. The TDNN model is given an initial input I_0, and it predicts the next state of the satellite which is fed back as the new input. Fig. 2.3 depicts the structure of TDNN.

LSTMs are mainly composed of a cell which remembers values over time intervals (memory

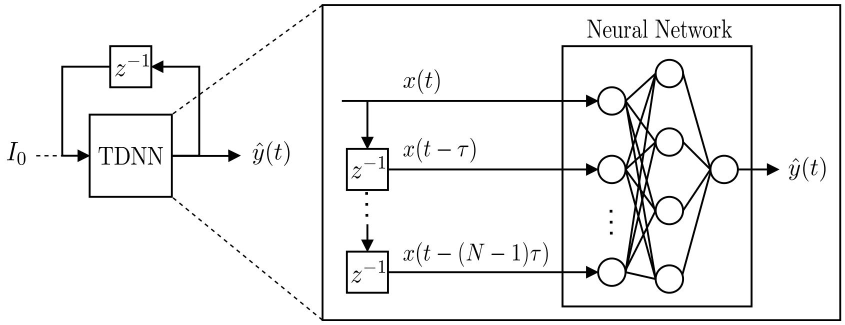


Figure 2.3: TDNN architecture.

cell), an input gate, output gate, and a forget gate which regulates the flow of information into and out of the cell [67]. Fig. 2.4 presents the structure of an LSTM NN.

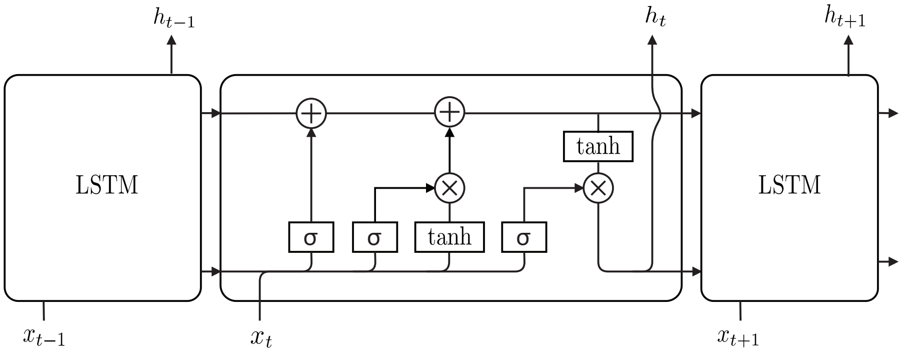


Figure 2.4: Generic LSTM Architecture.

The next subsection compares the performance of TDNN and LSTM for short orbital prediction.

Results

A simulation study was conducted using the data generated from the AGI-STK software corresponding to the an Orbcomm LEO satellite. Table 2.1 summarizes the simulation settings. Fig. 2.5 plots the training and validation MSEs as a function of training epochs. The validation errors for the TDNN design are low, showcasing its ability to function within an acceptable margin of error with new data. The LSTM is shown to be capable of following the profile of the LEO satellite's orbit, however, its accuracy was much lower.

Table 2.1: Simulation Settings.

Parameter	Value
Satellite name	Orbcomm F107
Duration [hours]	10
Sampling time [seconds]	1
Training Period [hours]	5
Testing Period [hours]	5

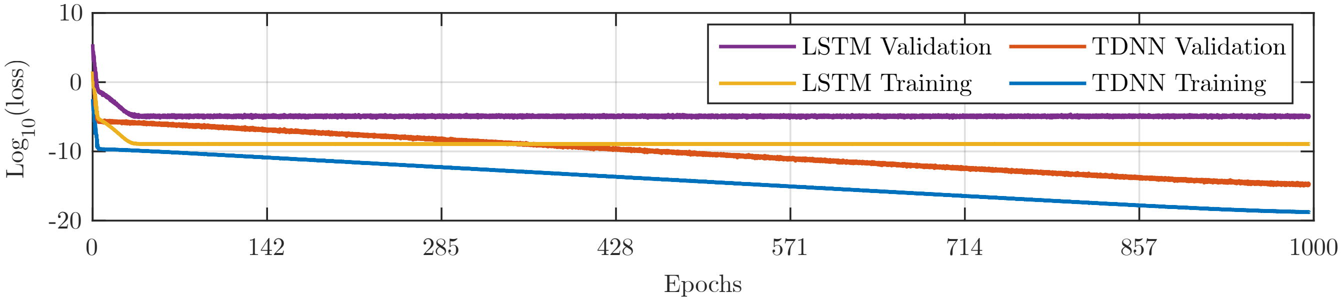


Figure 2.5: Validation loss (MSE) versus number of training epochs of proposed models.

The training and validation errors show that the TDNN architecture outperforms the LSTM. The TDNN model is assessed by predicting the satellite's ephemeris in a closed-loop fashion. A TDNN model was trained for every output pair of parameters in the orbital state $S = \{x, y, z, vx, vy, vz\}$. For each coordinate, a model was trained to output its value and derivative. Simulation results showed promising propagation accuracy for long-term windows using the proposed TDNN-based approach compared to the conventional SGP4 model. Fig. 2.6 depicts the error comparison of SPG4 and the TDNN corresponding to the $\{x, \dot{x}\}$ pair.

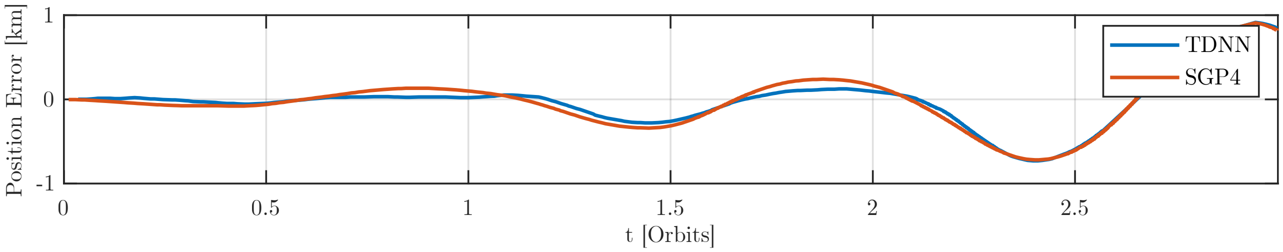


Figure 2.6: LEO satellite's position error comparison between SGP4 and TDNN.

The results suggest that the TDNN model is capable of following the HPOP propagator for a short time window. The TDNN showed an acceptable performance for about one orbit, after which the performance starts degrading.

2.4.2 Error Propagation

The errors of the proposed ANNs with respect to HPOP were close to those of SGP4. This could be attributed to the complexity of the LEO satellite's orbit, as well as the high dimensional properties of HPOP. SGP4 itself is accurate as well, though not as accurate as HPOP, in propagating the ephemeris. Therefore, a nonlinear autoregressive with exogenous inputs (NARX) model was devised to map the output of an SGP4 propagator to those of a well-initialized HPOP propagator.

Training

To compare both proposed frameworks, the same training, validation, and testing data extracted from AGI-STK's HPOP propagated ephemeris are utilized.

Design

The NARX model essentially functions similarly to a NAR model, with the only difference being that exogenous inputs, which are propagated SPG4 ephemeris, are utilized in predicting the output, as seen in Fig. 2.7. The NARX architecture has been shown to be highly capable of learning long-term dependencies [68] and predicting time series [69, 70, 71]; even chaotic time series [72].

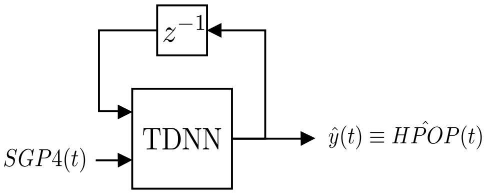


Figure 2.7: Proposed NARX model.

The NARX architecture takes SGP4 propagated position states as inputs. It also has a feedback loop where its output, the estimated HPOP state values, are fed back as additional input. This way, the model takes SGP4 as input as well as previous estimated $\hat{\text{HPOP}}(t-1,\ldots,t-d)$ values, and it outputs new estimated $\hat{\text{HPOP}}(t)$ values.

Results

The simulation settings shown in Table 2.1 for the Orbcomm LEO satellite were used. Fig. 2.8 and Table 2.2 show the position error magnitude of the SGP4 versus the propagated SGP4+NARX approach. Because the SGP4 propagator is initialized with TLE files, which are a compact means to achieve modestly fast and accurate calculation, its initial position error can reach up to 1 km [73, 74, 75]. Fig. 2.8 (a) shows the SGP4 position error magnitude while keeping the initial TLE error. Fig. 2.8 (b) shows the SGP4 position error magnitude with the initial error of the TLE file removed for a fairer comparison since both graphs would start from 0 error. The results show that SGP4+NARX is accumulating error at a much slower rate compared to SGP4.

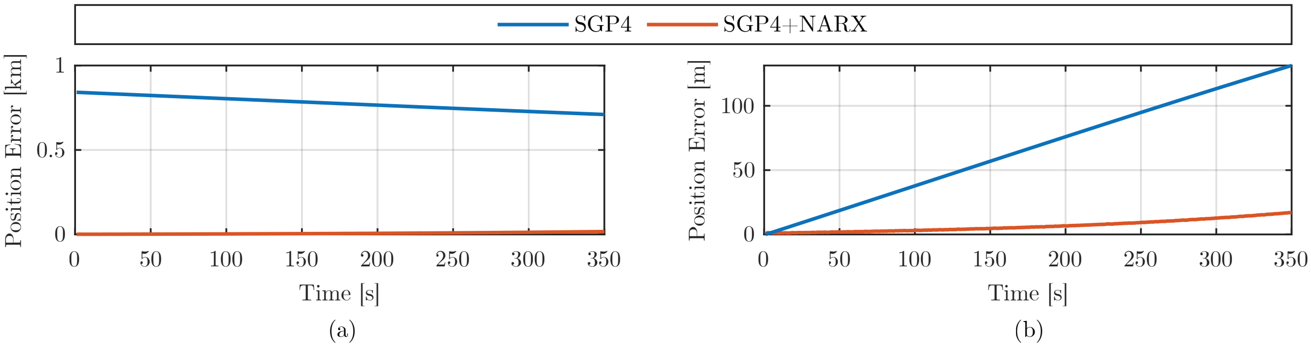


Figure 2.8: 3-D Position error magnitude for Orbcomm FM107's trajectory (a) without removing the initial error of SGP4 and (b) with removing the initial error of SGP4.

From these results, it can be concluded that SGP4+NARX offers promising orbit propagation.

Table 2.2: LEO propagation performance in simulation for 350 seconds.

RMSE [m]	**Orbcomm-FM107**
SGP4 (with TLE initial error)	773
SGP4 (without TLE initial error)	54
SGP4+NARX	8

2.5 Model Investigation

The chosen model's performance is investigated. First, to get the best of our proposed model, we make use of tuners that allow us to optimally look within a specified search space and get the best hyper-parameters. Figure 2.9 shows an overview of some tuners considered. Both the Hyper-band and Bayesian tuners were utilized and the hyper-parameters to which each tuner converged were investigated. Table 2.3 shows the hyper-parameter search space. Both tuners converged to the same hyper-parameters. It is important to note that adding multiple RNN layers or Dense layers does not lead to better results.

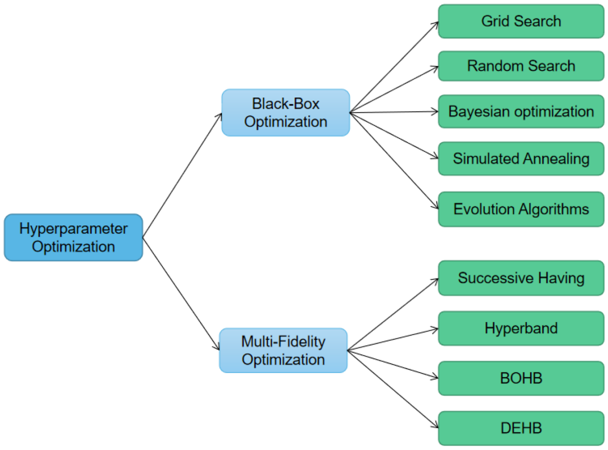


Figure 2.9: Overview of hyper-parameter tuners.

Figure 2.10 shows a sample of the progression of weights over training epochs for multiple models with different hyper-parameters. Finally, the tuners agreed on two hidden RNN layers with 2 Dense layers. The number of layers per node is 10 with no noticeable increase in performance when using a higher number of nodes. The optimizer used here is Yogi[78];

Table 2.3: Hyper-parameter Search Space.

Parameter	**Range**
Number of Layers	[1,10]
Number of nodes/layer	[4,512]
Input features	Any subset from x, y, z, v_x, v_y, v_z
Prediction Length	[0.1s, 4 minutes]
Optimizer	Adam[76], Adagrad[77], SGD, Yogi[78]

Adam[76] produces similar results while maintaining generality, but Yogi tends to get much closer to the validation data. The prediction length is set to 400 time delays where the time interval between delays is 0.01 seconds. It is interesting to compare the derived model to a less complex model. We utilize the Hyper-band tuner to give us the best hyper-parameters for a simple neural network using only Dense layers. The hyper-parameters tuned are the number of layers ([0,2] hidden layers), number of nodes per layer([20,420] nodes), activation function for each layer, and the initial learning rate. Figure 2.11 shows a sample of some neural network models trained over a specific search space. The lines each represent a unique model. A line specifies which hyper-parameters its model was built with and then lands on the derived values of the training loss and validation loss. The inputs are scaled with a "MinMax" scaler which squeezes the values of each state between 0 and 1. Since the states are input in meters, and the satellites are 700km in altitude, the values are scaled down to a minimum value in the order of 10^{-6}. With utilizing the "Mean Square Error" as our loss function, a meter-level solution means a loss of order less than $(10^{-6})^2 = 10^{-12}$. Figure 2.11 shows that even models with a linear activation function and no hidden units, can achieve meter-level accuracy for one-step prediction. This indicates that the neural network derives that to output the correct ephemeris, it would only need to output a weighted version of a time-shifted SGP4-propagated ephemeris. Therefore, one method for ephemeris propagation when given a sparse amount data relating to a satellite's true position is to find a time shift that matches the SGP4-propagated ephemeris with the satellite's true ephemeris and use that time-shifted ephemeris.

Figure 2.10: Progression of weights throughout epochs for multiple models.

2.6 Conclusion

This chapter gives a brief summary on machine learning in general and its usage in satellite ephemeris propagation. This can be utilized in multiple areas such as space situational

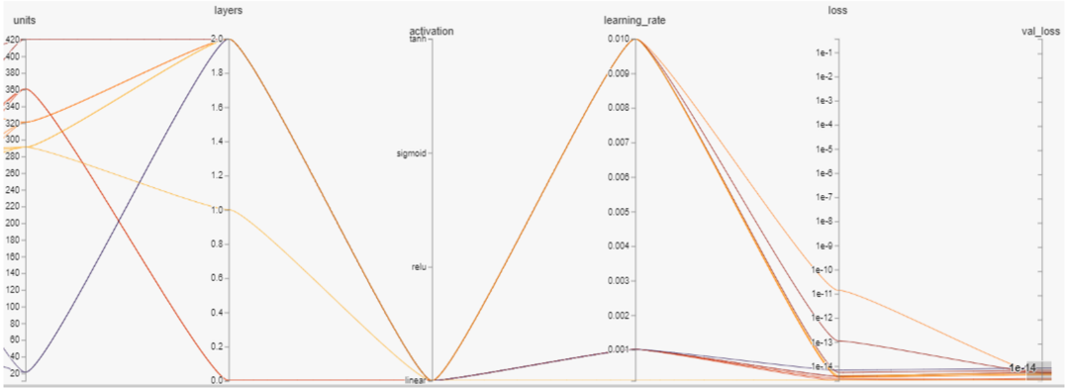


Figure 2.11: Sample results for Hyper-band tuning.

awareness, positioning, and navigation. Multiple models are investigated utilizing decoded LEO satellite ephemeris from Orbcomm satellites used as ground truth targets. The best model found is an autoregressive model that takes its past outputs as new inputs as well as past and future SGP4-propagated ephemeris that uses TLE files as for its initialization.

Chapter 3

Machine Learning aided Simultaneous Tracking and Navigation

3.1 STAN Overview

This section presents a high-level block diagram of the proposed system. The proposed system builds on the traditional STAN framework introduced in [9]. The performance of the STAN framework has been previously demonstrated in realistic simulation environments and experimentally on a ground vehicle and on an unmanned aerial vehicle (UAV), showing the potential of achieving meter-level-accurate navigation [51, 79, 9].

The STAN framework utilizes specialized LEO receivers to extract navigation observables, such as ephemeris messages, if available, pseudorange, and Doppler measurements from the LEO satellite signals. Furthermore, a model-based LEO propagator, such as "Two-Body with J2" initialized with SGP4-propagated ephemeris, is employed in estimating the LEO satellite's states (position and velocity). The ML-STAN framework instead replaces the model-based propagator with an ML-based propagator to achieve potentially more accurate

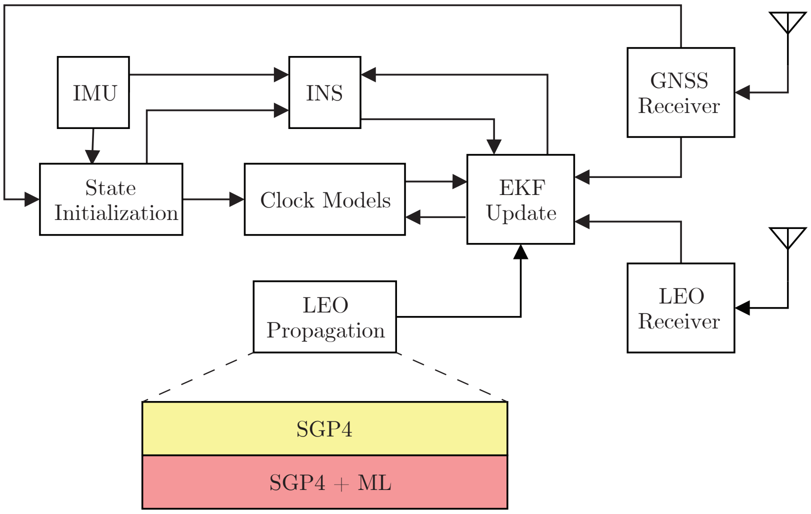


Figure 3.1: Proposed improvements on the STAN system.

propagated LEO states. The framework includes GNSS receivers, which produce a navigation solution from GNSS signals, when such signals are available. Finally, an IMU which reports the vehicle's specific force, angular rate, and orientation, is embedded within an INS. The INS provides the vehicle's position state, which along with the LEO signals, propagated LEO states, GNSS signals, and clock models which compensate for timing bias and phase shifts, are fed into an Extended Kalman Filter (EKF). The employed EKF then simultaneously estimates the vehicle's states, tracks the LEO satellite's states, and estimates timing biases as well as the confidence of the estimated values. When GNSS signals are available, the framework uses these signals for navigating the vehicle and tracking the LEO satellites. Once the GNSS signals cut off, the ML model propagates the LEO satellite states within the STAN framework. Fig. 3.1 depicts the proposed EKF-ML-STAN framework.

The proposed ML propagator approach could take one of two forms. The first form replaces the LEO propagation block with a specialized ML propagation block capable of directly estimating LEO satellite ephemeris. The second form adds an error correction block to the STAN framework right after the LEO propagation block. The error correction block learns the mapping between a standard propagator, such as SGP4, and a highly accurate propagator, such as HPOP, that is fitted onto ground truth ephemeris transmitted by LEO satellites.

3.2 Experimental Results

In this section, the performance of the proposed STAN navigation framework is assessed experimentally. Here, the performance utilizing propagation via Two Body with J2, initialized with SPG4-propagated ephemeris, will be compared to the one achieved with the proposed SGP4+NARX. In the experiment, Orbcomm LEO satellites signals were collected, where a ground-truth reference of the satellite ephemeris was obtained by decoding the satellites' positions transmitted from their on-board GPS receivers [80]. The error estimation NARX architecture was chosen, as it has demonstrated the highest accuracy among the studied ML frameworks.

3.2.1 Environmental Layout and Experimental Setup

A ground vehicle was equipped with the following hardware and software setup:

- A quadrifilar helix antenna to receive the Orbcomm SV downlink signals, which are transmitted at frequencies between 137 and 138 MHz
- A USRP E312 to sample Orbcomm symmetric differential phase shift keying (SDPSK) signals.
- These samples were then processed by the Multi-channel Adaptive TRansceiver Information eXtractor (MATRIX) software-defined radio developed by the Autonomous Systems Perception, Intelligence, and Navigation (ASPIN) Laboratory to perform carrier synchronization, extract pseudorange rate observables, and decode Orbcomm ephemeris messages [80].

- A Septentrio AsteRx-i V integrated GNSS-IMU, which is equipped with a dual-antenna, multi-frequency GNSS receiver and a Vectornav VN-100 micro-electromechanical system (MEMS) IMU. Septentrio's post-processing software development kit (PP-SDK) was used to process GPS carrier phase observables collected by the AsteRxi V and by a nearby differential GPS base station to obtain a carrier phase-based navigation solution. This integrated GNSS-IMU real-time kinematic (RTK) system [81] was used to produce the ground truth results with which the proposed navigation framework was compared.

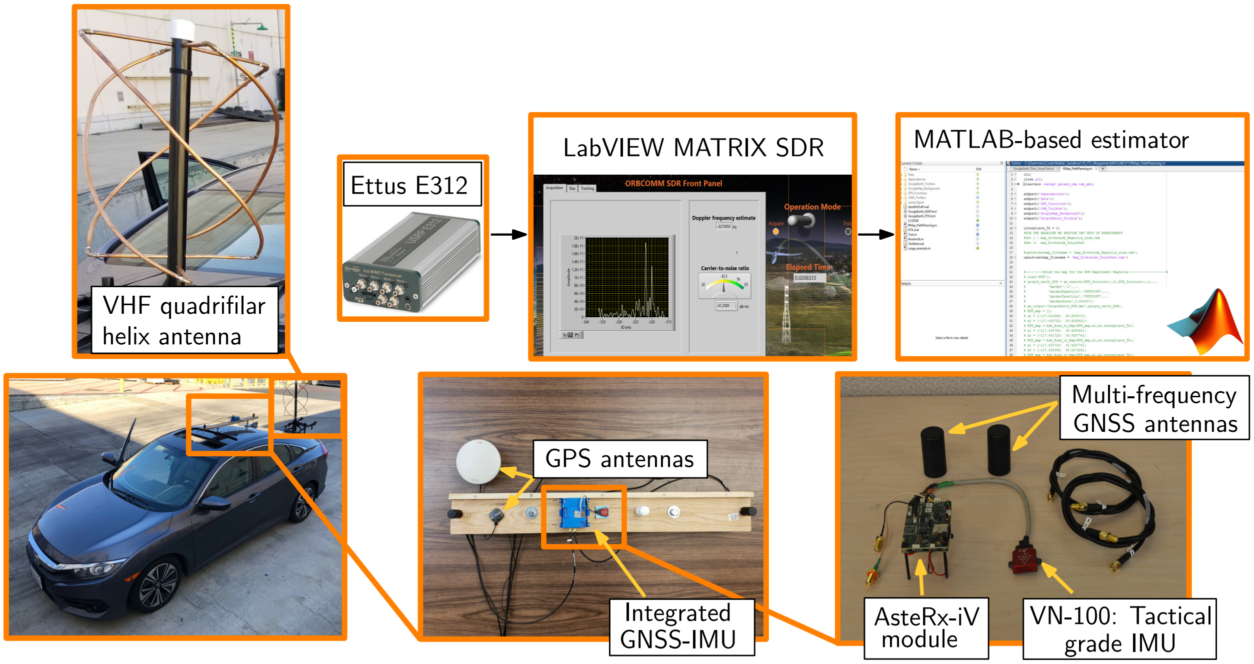


Figure 3.2: Experimental Setup.

The experimental setup is shown in Fig. 3.2. The ground vehicle was driven along U.S. Interstate 5 near Irvine, California, USA, for 7,495 m over 258 seconds, during which 2 Orbcomm LEO satellites were available. The standard deviation of the Orbcomm Doppler measurements was set to be 4.7 Hz, which was obtained empirically. Three navigation frameworks were implemented to estimate the vehicle's trajectory: (i) the LEO signal-aided INS STAN framework using Two body with J2 initialized with SGP4-propagated ephemeris as its propagator, (ii) the LEO signal-aided INS STAN framework using the NARX as its propagator,

Table 3.1: LEO Propagation Performance in an experimental scenario.

RMSE [m]	**Orbcomm FM112**	**Orbcomm FM117**
SGP4	558	1,226
SGP4+NARX	74	38

(iii) a traditional GPS-aided INS for comparative analysis. The SGP4 propagated ephemeris was initialized from the corresponding TLE file published by NORAD. The HPOP ephemeris utilized for training was properly initialized through STK, using the Orbcomm LEO satellite's positions decoded from the first measurement epoch. To perform a fair comparison between the propagation of SGP4+NARX that uses the first decoded ephemeris point for initialization and the performance of SGP4 that uses a relatively old TLE file, we generated a new TLE file from the transmitted ephemeris, propagated it with SGP4, and used it for the comparison. Fig. 3.3 sketches the data generation and prediction phases.

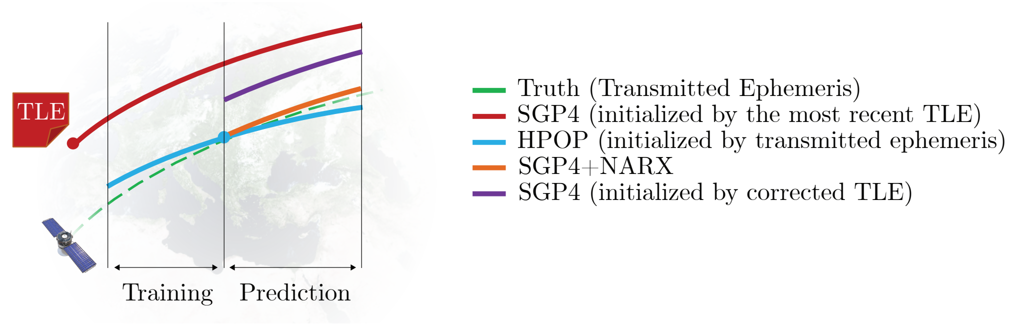

TLE
Training
Prediction
🟩 Truth (Transmitted Ephemeris)
🟥 SGP4 (initialized by the most recent TLE)
HPOP (initialized by transmitted ephemeris)
🟧 SGP4+NARX
🟪 SGP4 (initialized by corrected TLE)

Figure 3.3: Sketch illustrating the training phase of the NARX model (using TLE-SGP4 and HPOP) as its prediction phase.

3.2.2 Results

First, the performance of the SGP4+NARX model as a propagator is compared with SGP4. The ephemeris position error $E_{\text{LEO}} = \{E_x, E_y, E_z\}$ of both tracked satellites and the total positional error characterized by $E_r = \sqrt{E_x^2 + E_y^2 + E_z^2}$ are shown in Fig. 3.4 and summarized in Table 3.1.

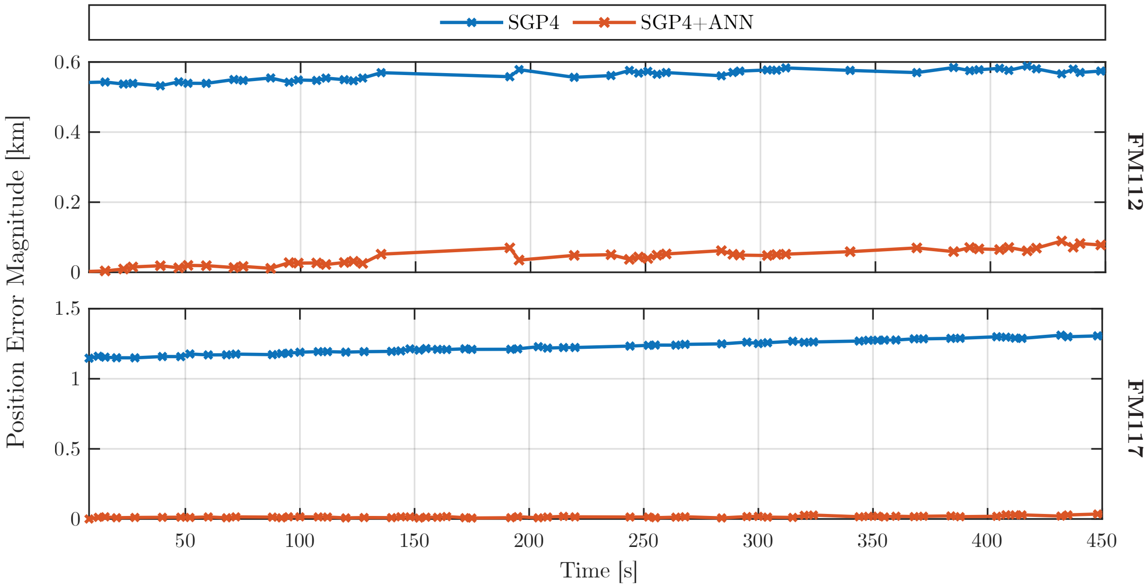


Figure 3.4: Comparison between SGP4 and SGP4+NARX propagation

The improved results in LEO satellite position estimation translate directly to a better navigation performance for a ground vehicle. Fig. 3.5 shows the vehicle's true trajectory compared to estimates from the original STAN framework and the proposed STAN with SGP4+NARX. The results are summarized in Table 3.2. The results show how the NARX model's ability in better estimating the LEO satellites' ephemeris leads to a more accurate navigation solution.

Table 3.2: Ground Vehicle Navigation Performance.

Performance Measure	**STAN with SGP4**	**STAN with SGP4+NARX**
RMSE [m]	30	3.6
Final Error [m]	30	8.3

3.3 Conclusion

An experiment was conducted with a ground vehicle navigating while measurements were extracted from two Orbcomm LEO satellites. The training data for the ML model was historical HPOP ephemeris data of the satellite's orbit. The performance of the SGP4+NARX

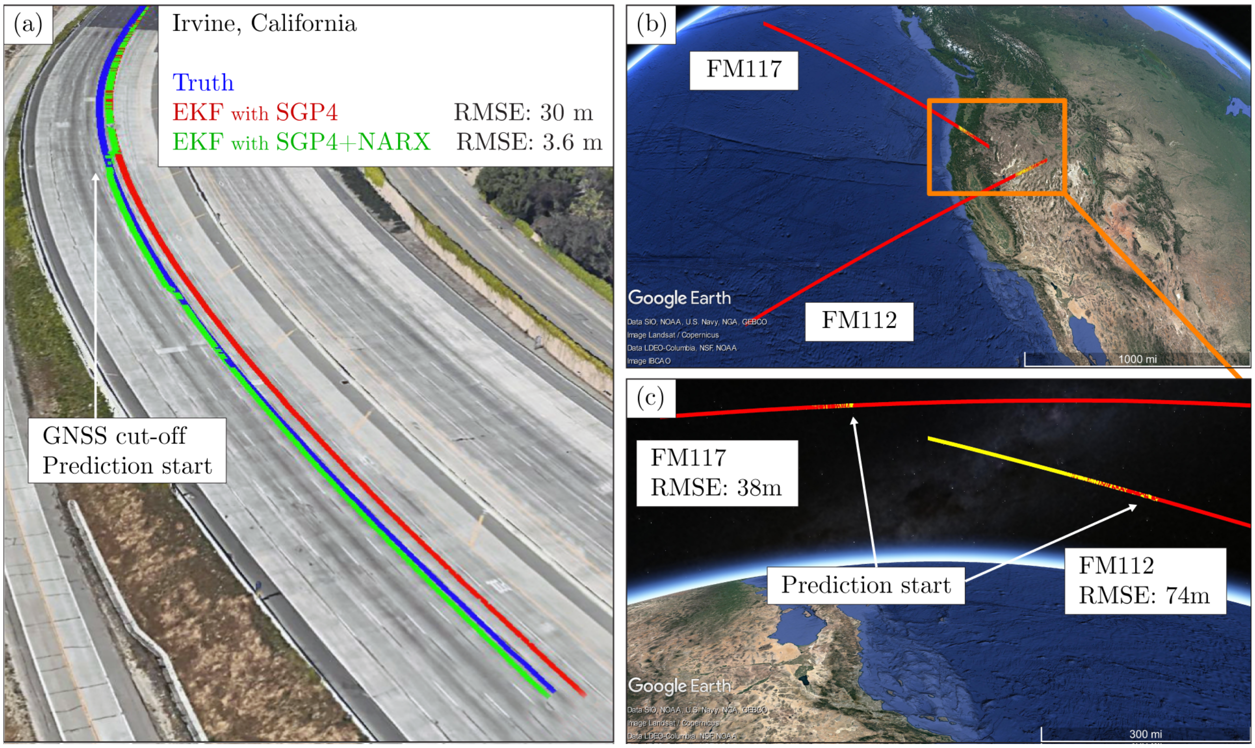


Figure 3.5: Experimental results showing (a) the trajectory of a ground vehicle navigating with the proposed approach. The truth (white) is compared to the STAN with SGP4 estimate (red) and the STAN with SGP4+NARX estimate (green). (b, c) the trajectory of the 2 Orbcomm LEO satellites generated by SGP4+NARX predictions (yellow) versus the truth trajectories (red) obtained from onboard GPS receivers. Map data: Google Earth.

model in tracking the LEO satellites' ephemeris was compared with the results of a traditional LEO propagation model using SPG4. The EKF with SGP4+NARX framework noticeably outperformed the traditional EKF with SGP4 framework's accuracy. The STAN with SGP4 achieved a ground vehicle 3-D position RMSE of 30 m. In contrast, the proposed STAN with SGP4+NARX framework achieved a ground vehicle 3-D position RMSE of 3.6 m, while the two LEO satellites were tracked with 3-D position RMSE of 38 m and 74 m.

Chapter 4

Hybrid Framework for Unavailable Target Ephemeris

In the previous chapter, transmitted satellite ephemerides were used as ground truth target values to train our machine learning model. However, it is not the case that LEO satellites publicly transmit their live ephemeris. Therefore, a new approach must be proposed to get the target values which the model will train on for satellite ephemeris prediction. This chapter proposes a hybrid numerical-machine learning approach for LEO satellite orbit prediction where the receiver has no prior knowledge on the satellite's position except for publicly available TLE files. The following contributions are made:

- A hybrid numerical-ML propagator is developed in a three-step framework: (i) refining a LEO satellite's ephemeris via opportunistic tracking initialized from an SGP4-propagated TLE, (ii) training the ML propagator on the refined ephemeris, thus extending the approach in [51, 52] to no prior true ephemeris information by solely relying on initialization from TLEs, and (iii) localizing a receiver opportunistically with the ML-propagated ephemeris.

- The ephemeris propagation performance of the hybrid numerical-ML framework is compared with that of standalone propagators using true decoded Orbcomm ephemeris data.

- Experimental results are presented to demonstrate the efficacy of the proposed framework with a first pass of an Orbcomm satellite during which the tracking to refine the ephemeris is performed and a second pass of the same satellite during which the opportunistic localization performance using ML-propagated ephemeris and SGP4-propagated ephemeris are compared.

4.1 Carrier-Phase Measurement Model

Unlike GNSS satellites that periodically transmit accurate ephemeris information and clock corrections in their navigation message, LEO satellites do not openly transmit such information in general. Therefore, carrier-phase measurements are extracted opportunistically from LEO satellites to allow for tracking and localization. The following section describes the opportunistic carrier-phase measurement model.

A LEO receiver extracts continuous-time carrier phase measurements from LEO satellites' signals by integrating the Doppler measurement over time [82]. The carrier phase measurement (expressed in meters) made by the receiver on the LEO satellite at time-step k, which represents discrete-time instant $t_k = kT + t_0$ for an initial time t_0, can be modeled in discrete-time as

$$\begin{aligned}\phi(k) &= \left\|\boldsymbol{r}_{\mathrm{r}}(k) - \boldsymbol{r}_{\mathrm{leo}}(k')\right\|_2 + c\left[\delta t_r(k) - \delta t_{\mathrm{leo}}(k'_l)\right] + \lambda N \\ &\quad + c\delta t_{\mathrm{iono}}(k) + c\delta t_{\mathrm{tropo}}(k) + v(k), \quad k = 1, 2, \ldots \end{aligned} \tag{4.1}$$

where k' represents discrete-time at $t_{k'} = kT + t_0 - \delta t_{\mathrm{TOF}}$, with δt_{TOF} being the true time-of-flight of the signal from the LEO satellite to the receiver, $\boldsymbol{r}_r$ and $\boldsymbol{r}_{\text{leo}}$ are the receiver's and LEO satellite's 3-D position vectors expressed in the same reference frame, λ is the wavelength of the carrier signal transmitted by the LEO satellite, N is the carrier phase ambiguity of the LEO satellite carrier phase measurement, δt_{iono} and δt_{tropo} are the ionospheric and tropospheric delays, respectively, affecting the LEO satellite's signal, and v is the measurement noise, which is modeled as a zero-mean white Gaussian random sequence with variance σ_ϕ^2.

Assuming no cycle slip occurs when the receiver tracks the carrier phase (i.e., the carrier phase ambiguity remains constant), the difference between the receiver and the LEO satellite clock biases and the carrier phase ambiguity are lumped into a single term $c\Delta\delta t(k)$ to simplify the carrier phase measurement model between the receiver and the satellite in (4.1) into

$$\phi(k) = \|\boldsymbol{r}_r(k) - \boldsymbol{r}_{\text{leo}}(k')\|_2 + c\Delta\delta t(k) + c\delta t_{\text{tropo}}(k) - c\delta t_{\text{iono}}(k) + v(k), \tag{4.2}$$

$$c\Delta\delta t(k) \triangleq c\left[\delta t_r(k) - \delta t_{\text{sv}}(k')\right] + \lambda N. \tag{4.3}$$

4.2 Orbital Determination Framework

Localization with carrier phase measurements from an overhead satellite requires knowledge of the satellite's ephemeris. The best estimate available publicly for a LEO satellite's ephemeris would be from an initial ephemeris at a given time provided by the TLE files published by NORAD, which would then be propagated through SGP4. However, this initial ephemeris could be off by a few kilometers, and with SGP4 propagating the satellite's ephemeris through time, the error would continue to increase, only to be corrected when a new TLE file is published the next day. This is not ideal for use in localization where using wrong ephemeris as-is in the localization filter could yield huge localization errors.

The proposed framework mitigates the issue of relying on an ephemeris that is off by several kilometers. A base station with known position opportunistically tracks and refines the satellite's ephemeris when it is overhead. Next, a neural network trains on this refined ephemeris and predicts the satellite's future position and velocity. Finally, a receiver with unknown position is given this predicted ephemeris and uses this ephemeris instead of the TLE one in localizing itself when the same satellite passes overhead. Figure 4.1 illustrates the proposed framework. The following subsections describe each stage of the proposed framework.

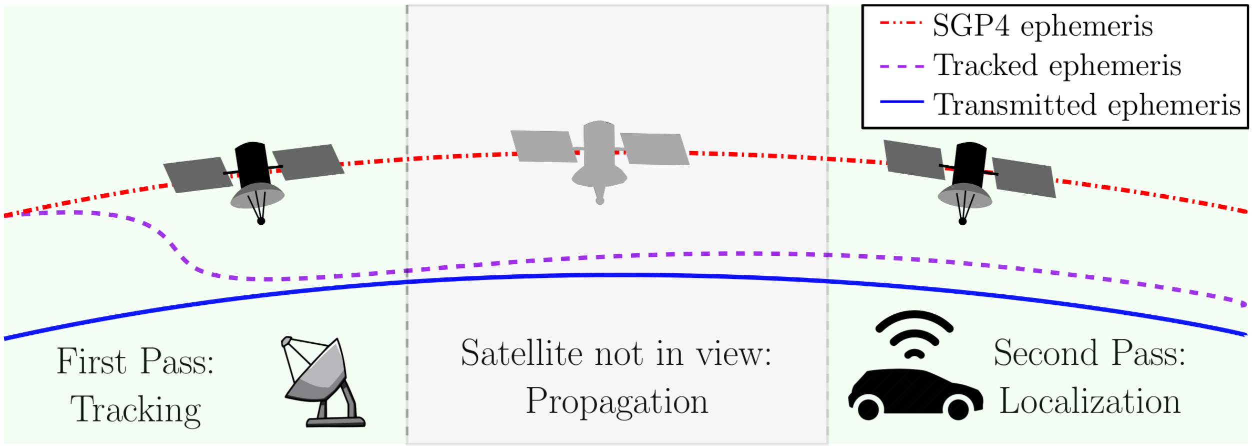


Figure 4.1: Proposed Framework.

4.2.1 Opportunistic Tracking

This subsection formulates the extended Kalman filter (EKF) utilized as the tracking filter used to estimate the LEO satellite's ephemeris in the first stage of the proposed framework. The filter assumes a base station with known position receiving carrier phase measurements modeled in (4.1). The EKF state vector is given by

$$\boldsymbol{x}_{\text{leo}} = \left[\boldsymbol{r}_{\text{leo}}^{\mathsf{T}},\, \dot{\boldsymbol{r}}_{\text{leo}}^{\mathsf{T}},\, c\Delta\delta t,\, c\Delta\dot{\delta}t\right]^{\mathsf{T}}$$

where the $\boldsymbol{x}_{\text{leo}}$ is the LEO satellite state vector, composed of the LEO satellite's position $\boldsymbol{r}_{\text{leo}}$ and velocity $\dot{\boldsymbol{r}}_{\text{leo}}$ vectors, expressed in the Earth-centered inertial (ECI) frame, and the lumped bias term, $c\Delta\delta t$, and drift term, $c\Delta\dot{\delta}t$, are those described in (4.3). The model dynamics of the clock terms are described in Appendix 7.5.

For the satellite's time update between measurements, a two-body model including the most significant non-zero mean perturbing acceleration, which corresponds to J_2 effects, is adopted as the LEO satellite orbital dynamics model in the ECI reference frame. This model offers a trade-off between accurate open-loop state prediction while maintaining a simple analytical Jacobian for estimation error covariance propagation and is given as follows

$$\ddot{\boldsymbol{r}}_{\text{leo}} = \boldsymbol{a}_{\text{grav},J_2} + \tilde{\boldsymbol{w}}_{\text{leo}}, \qquad \boldsymbol{a}_{\text{grav},J_2} = \frac{dU_{J_2}}{d\boldsymbol{r}_{\text{leo}}}, \tag{4.4}$$

where $\boldsymbol{r}_{\text{leo}} \triangleq [x_{\text{leo}},\, y_{\text{leo}},\, z_{\text{leo}}]^{\mathsf{T}}$ is the 3-D position vector of the LEO satellite in the ECI frame, $\boldsymbol{a}_{\text{grav},J_2}$ is the acceleration due to Earth's non-uniform gravity including J_2 effects, U_{J_2} is the non-uniform gravity potential of Earth including J_2 effects at the satellite, and $\tilde{\boldsymbol{w}}_{\text{leo}}$ is a process noise vector with power spectral density (PSD) $\tilde{\mathbf{Q}}_{\text{leo}}$, which attempts to capture the overall acceleration perturbations including the unmodeled non-uniformity of Earth's gravitational field, atmospheric drag, solar radiation pressure, third-body gravitational forces (e.g., gravity of the Moon and Sun), and general relativity [35].

The two-body with J_2 model, which allows for accuracies of a few meters in a short period of time, describes the components of $\boldsymbol{a}_{\text{grav},J_2} = [\ddot{x}_{\text{grav}},\, \ddot{y}_{\text{grav}},\, \ddot{z}_{\text{grav}}]^{\mathsf{T}}$ as

$$\begin{aligned}
\ddot{x}_{\text{grav}} &= -\frac{\mu x_{\text{leo}}}{\|\boldsymbol{r}_{\text{leo}}\|^3}\left[1 + J_2\frac{3}{2}\left(\frac{R_E}{\|\boldsymbol{r}_{\text{leo}}\|}\right)^2\left(1 - 5\frac{z_{\text{leo}}^2}{\|\boldsymbol{r}_{\text{leo}}\|^2}\right)\right],\\
\ddot{y}_{\text{grav}} &= -\frac{\mu y_{\text{leo}}}{\|\boldsymbol{r}_{\text{leo}}\|^3}\left[1 + J_2\frac{3}{2}\left(\frac{R_E}{\|\boldsymbol{r}_{\text{leo}}\|}\right)^2\left(1 - 5\frac{z_{\text{leo}}^2}{\|\boldsymbol{r}_{\text{leo}}\|^2}\right)\right],\\
\ddot{z}_{\text{grav}} &= -\frac{\mu z_{\text{leo}}}{\|\boldsymbol{r}_{\text{leo}}\|^3}\left[1 + J_2\frac{3}{2}\left(\frac{R_E}{\|\boldsymbol{r}_{\text{leo}}\|}\right)^2\left(3 - 5\frac{z_{\text{leo}}^2}{\|\boldsymbol{r}_{\text{leo}}\|^2}\right)\right],
\end{aligned} \tag{4.5}$$

where μ is Earth's standard gravitational parameter, and R_E is the mean radius of the Earth.

4.2.2 Localization

This subsection formulates the Extended Kalman Filter (EKF) utilized as the localization filter used to estimate the receiver's position in the second stage of the proposed framework. The filter assumes a stationary receiver with unknown position and clocks. The receiver opportunistically extracts carrier phase measurements as modeled in 4.1. The EKF state vector is given by

$$\boldsymbol{x}_{\mathrm{r}} = \left[\boldsymbol{r}_{\mathrm{r}}^{\mathsf{T}}, \, c\Delta\delta t_{\mathrm{r}}, \, c\Delta\dot{\delta} t_{\mathrm{r}} \right]^{\mathsf{T}}$$

where the $\boldsymbol{x}_{\mathrm{r}}$ is the receiver state vector composed of the receiver's position $\boldsymbol{r}_{\mathrm{r}}$ vector, expressed in the ECEF frame, and the lumped clock terms follow the same model as those described in section 4.2.1. The receiver transition matrix is $\mathbf{F}_r = \mathbb{I}_{3\times 3}$ since the receiver is stationary.

4.3 Experimental Results

In this section, the proposed framework is experimentally demonstrated by localizing a stationary receiver and comparing the performance of a filter using the refined ML-propagated ephemeris against using SGP4-propagated ephemeris initialized from TLE. In this experiment, an Orbcomm satellite was chosen since it is the only constellation that transmits the satellite's ephemeris publicly. Signals from satellite Orbcomm FM107 were collected, and using a specialized software defined radio, carrier-phase measurements were opportunistically extracted. Furthermore, the satellite's downlink signals, which include the satellite's true ephemeris generated by on-board GPS receivers, are decoded for comparison and used as ground truth. Finally, Ionospheric and Tropospheric delays are corrected for in the carrier-phase measurements [26].

4.3.1 Experimental Setup

A very high frequency (VHF) quadrifilar helix antenna was connected to an Ettus E312 Universal Software Radio Peripheral (USRP) to sample Orbcomm LEO satellites' signals at 137-138 MHz at a sampling rate of 2.4 MSps. The USRP's oscillator was disciplined by an external, freely-running NI CDA-2990 OctoClock. The recording was repeated twice to record two consecutive passes of Orbcomm FM107 LEO satellite over Irvine, California. The measurements opportunistically extracted from the LEO satellite's signals during the first pass are used to opportunistically track Orbcomm FM107 satellite's and refine its ephemeris. Then, this refined ephemeris is used as an input to the supervised training of the hybrid numerical-ML propagator. This propagator is subsequently employed to propagate the ephemeris of Orbcomm FM107 satellite's for around 100 minutes, which corresponds to its orbital period, until the satellite is visible again over Irvine. During the second pass, the measurements extracted from Orbcomm FM107 satellite's signals are used to opportunistically localize the receiver while using the hybrid numerical-ML propagated ephemeris. The localization performance is compared to that obtained using the SGP4-propagated TLE ephemeris. The skyplot showing the trajectory of the satellite for both consecutive passes is shown in Fig. 4.2.

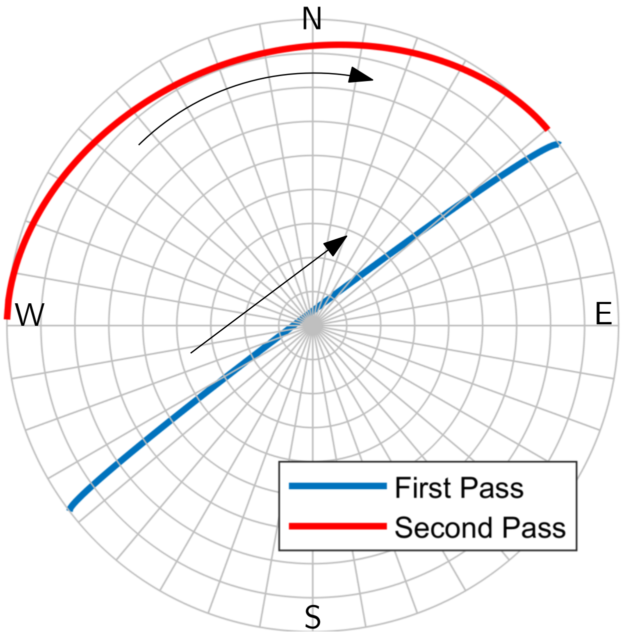


Figure 4.2: Skyplot of satellite Orbcomm FM107 during the tracking experiment (first pass) and the localization experiment (second pass).

4.3.2 Tracking

First, a base station with known position starts opportunistically extracting carrier-phase measurements to the satellite. The framework described in 4.2.1 is utilized in tracking the satellite and refining its ephemeris. The state estimates are initialized as follows:

$$\hat{\boldsymbol{x}}_{\text{leo}}(0|0) = [\boldsymbol{r}_{\text{leo}}^{\mathsf{T}}(0), \dot{\boldsymbol{r}}_{\text{leo}}^{\mathsf{T}}(0), \boldsymbol{z}(0) - \|\boldsymbol{r}_r - \boldsymbol{r}_{\text{leo}}(0)\|_2\,, 0]^{\mathsf{T}}$$

where $[\boldsymbol{r}_{\text{leo}}^{\mathsf{T}}(0), \dot{\boldsymbol{r}}_{\text{leo}}^{\mathsf{T}}(0)]^{\mathsf{T}}$ is the satellite's initial state given from the SGP4-propagated ephemeris initialized from TLE, in the ECI frame. The estimate of the lumped clock bias, $c\Delta\delta t$, is initialized as the difference between the initial measurement and the range from the base station to the satellite as propagated from TLE, and the drift, $c\Delta\dot{\delta}t$, is initialized at 0. The initial error estimation covariance was computed according to

$$
\begin{aligned}
&\boldsymbol{P}_{\boldsymbol{x}_{leo}}(0|0) \triangleq \mathrm{diag}[\boldsymbol{P}_i(0|0), \boldsymbol{P}_{\mathrm{clk}}(0|0)] \\
&\boldsymbol{P}_i(0|0) \equiv {}_b^i\bar{\boldsymbol{R}}(0)\boldsymbol{P}_b(0|0)\,{}_b^i\bar{\boldsymbol{R}}^{\mathsf{T}}(0) \\
&{}_b^i\bar{\boldsymbol{R}}(0) = \mathrm{diag}[{}_b^i\boldsymbol{R}(0), {}_b^i\boldsymbol{R}(0)] \\
&\boldsymbol{P}_b(0|0) \equiv \mathrm{diag}[5\times 10^5, 3\times 10^3, 10^5, 0.05, 0.01, 0.2] \\
&\boldsymbol{P}_{\mathrm{clk}}(0|0) \equiv \mathrm{diag}[100, 10]
\end{aligned}
$$

where $\boldsymbol{P}_i$ and $\boldsymbol{P}_b$ are the initial error covariance in the ECI frame and the satellite's body frame respectively, and ${}_b^i\boldsymbol{R}$ is the rotation matrix from the body frame to the ECI frame. This method initializes the estimate error covariance in the body frame which is more intuitive than initializing in the ECI frame, as it has been shown that most of the error is usually in the along-track. This allows to capture the elliptical nature of the error covariance and initializes cross terms in the ECI frame which would allow for faster and better convergence. Finally, the measurement noise covariance was set to 0.5 m^2. The satellite was tracked for 517 seconds, decreasing the magnitude of error in the satellite position from 980 m initially to 56 m with reference to the transmitted ephemeris. Figs. 4.3 and 4.4 show the EKF tracking plots for the satellite's ephemeris in the body frame. The cross track direction is the least observable, which is consistent with [83]. Fig. 4.5 shows the position error magnitude of the tracked ephemeris compared to that of the SGP4 ephemeris propagated from TLE.

4.3.3 Propagation

After tracking is finalized, the estimated ephemeris at the final time of tracking is considered to be the best estimate for the satellite's ephemeris. This ephemeris is then back-propagated using the J_2 orbital propagation model to smooth over the tracking period, since the J_2 orbital propagation model is accurate for short windows of time. This smoothed ephemeris

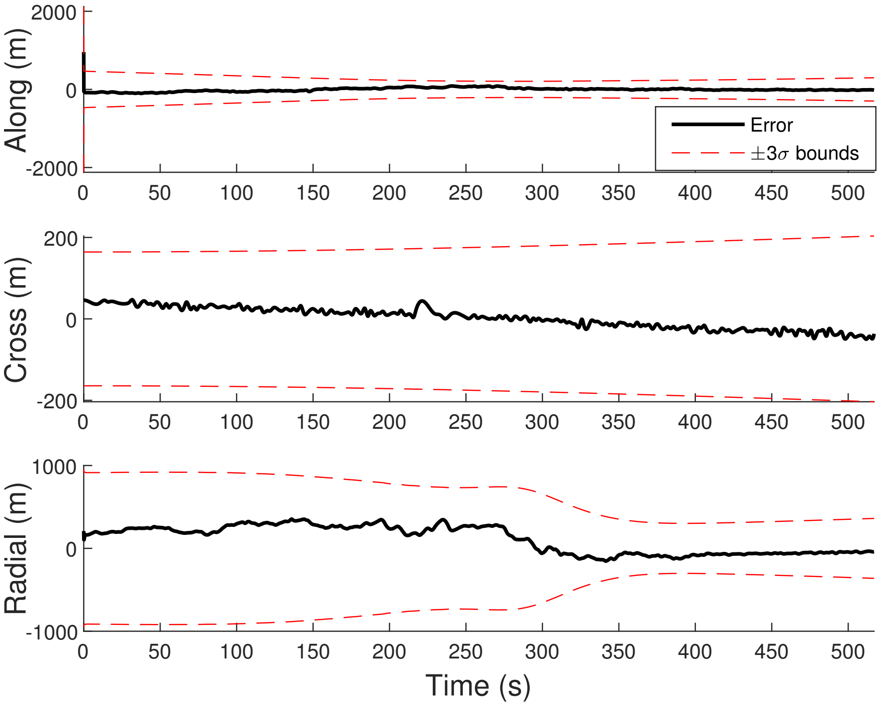


Figure 4.3: EKF position plots in the satellite's body frame.

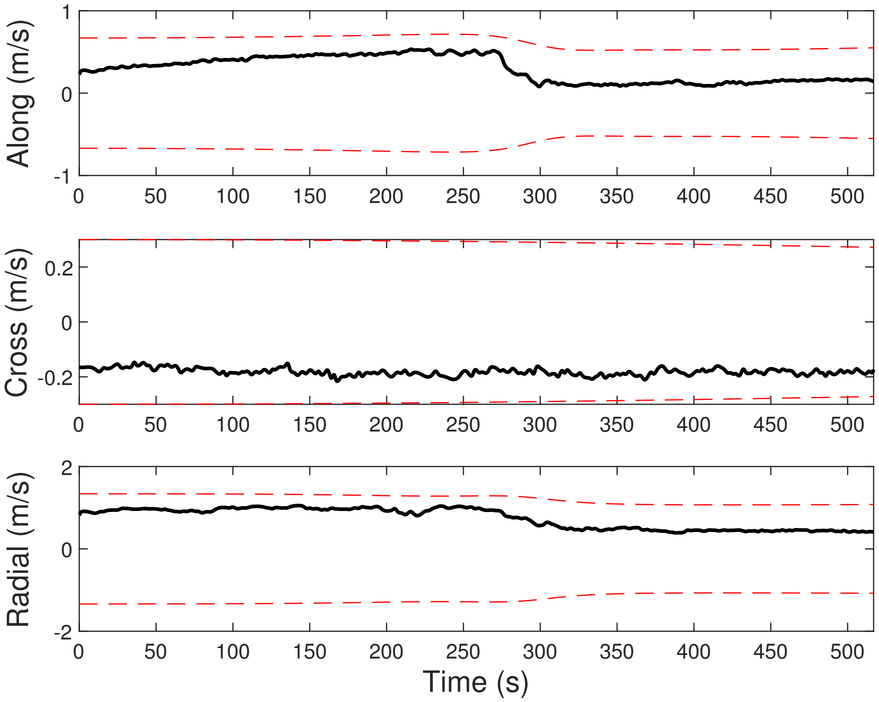


Figure 4.4: EKF velocity plots in the satellite's body frame.

is now ready to used for training. The ML model is given the continued SGP4 ephemeris propagated from TLE as exogenous input and the tracked and smoothed ephemeris as ground

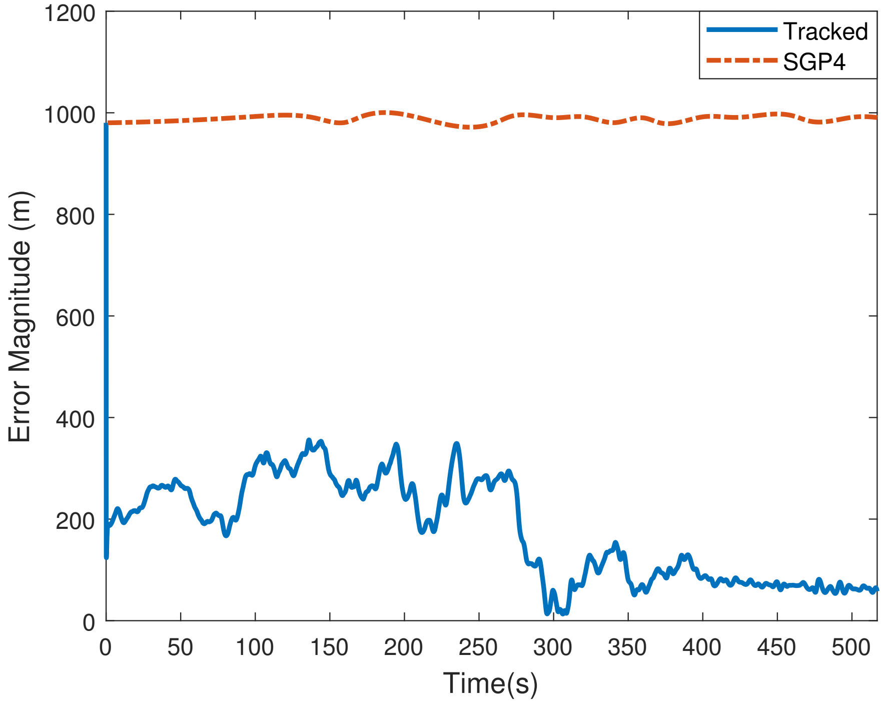


Figure 4.5: Position error magnitude comparison between SGP4 ephemeris initialized from TLE and tracked ephemeris.

truth. The ML model then trains on this data and learns a mapping from the SGP4-TLE ephemeris to the tracked ephemeris. Finally, the ML model starts propagating and extrapolating satellite ephemeris, taking as its input the SGP4-TLE ephemeris at each time step and its own outputs at previous time steps, and outputting a corrected ephemeris at that time step. The ML model is propagated for 5,870 seconds until the satellite comes back to view. The ephemeris is then further propagated a further period of 363 seconds for this ephemeris to be used in localization. Fig. 4.6 shows the magnitude of error for the ML-propagated ephemeris and that of the SGP4-propagated ephemeris when compared with the transmitted ephemeris during the second pass.

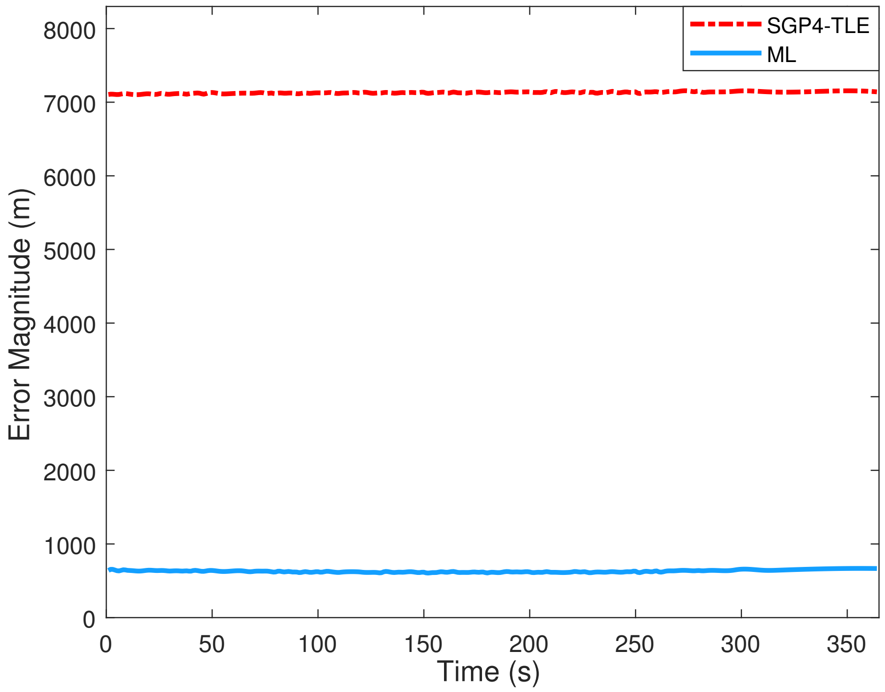


Figure 4.6: Position error magnitude comparison between SGP4 ephemeris initialized from TLE and the propagated ephemeris through the ML model.

4.3.4 Localization Results

Finally, the localization filter described in Subsection 4.2.2 is utilized for localizing a ground stationary receiver. The state estimates are initialized as follows

$$\hat{\boldsymbol{x}}_{\mathrm{r}}(0|0) = [\hat{\boldsymbol{r}}_{\mathrm{r}}^{\mathsf{T}}(0), \boldsymbol{z}(0) - \left\|\hat{\boldsymbol{r}}_r - \boldsymbol{r}_{\mathrm{leo}}^{j}(0)\right\|_2, 0]^{\mathsf{T}}$$

where $\hat{\boldsymbol{r}}_{\mathrm{r}}^{\mathsf{T}}(0)$ is the receiver's initial state which randomly drawn according to $\hat{\boldsymbol{r}}_{\mathrm{r}}^{\mathsf{T}}(0) \sim \mathcal{N}[\boldsymbol{r}_r, \boldsymbol{P}_e(0|0)]$, in the ECEF frame. The estimate of the lumped clock bias $c\Delta\delta t$ is initialized as the difference between the initial measurement and the estimated initial range where $j \in \{\mathrm{ML}, \mathrm{SGP4}\}$ is the index of the satellite ephemeris used. The drift $c\Delta\dot{\delta}t$ is initialized at 0. The initial error covariance matrices are as follows

$$\begin{aligned}
&\boldsymbol{P}_{\boldsymbol{x}_r}(0|0) \triangleq \text{diag}[\boldsymbol{P}_e(0|0), \boldsymbol{P}_{\text{clk}}(0|0)]\\
&\boldsymbol{P}_e(0|0) \equiv {}^e_l\bar{\boldsymbol{R}}\boldsymbol{P}_l(0|0)\,{}^e_l\bar{\boldsymbol{R}}^\mathsf{T}\\
&\boldsymbol{P}_l(0|0) \equiv \text{diag}[10^6, 10^6, 0.1]\\
&\boldsymbol{P}_{\text{clk}}(0|0) \equiv \text{diag}[10^8, 10^4]
\end{aligned}$$

where $\boldsymbol{P}_e(0|0)$ and $\boldsymbol{P}_l(0|0)$ are the initial receiver position error covariance in the ECEF frame and the local East-North-Up (ENU) frame respectively, and ${}^e_l\boldsymbol{R}$ is the rotation matrix from the local ENU frame to the ECEF frame. This allows for comparing with the two-dimensional (2-D) North-East error only since the Up direction is poorly estimable from satellite measurements only. Finally, the measurement noise covariance was set to 0.5 m^2. The receiver was localized for 363 seconds, using the ML-propagated ephemeris as the satellite's ephemeris in the EKF filter. The 2-D magnitude of error decreases from 2,219 m initially to 448 m with reference to the true receiver location. However, when the SGP4-propagated ephemeris, which is off by about 7 km, is utilized with the same measurement noise covariance, the error increases up to 6,718 m. This is to be expected as the EKF trusts the given ephemeris and attempts to decrease the measurement error covariance by positioning the receiver somewhere where the given measurements and given satellite ephemeris match. Fig. 4.7 shows the EKF localization plots for the receiver's position in the ENU frame. Fig. 4.8 shows the experimental results with the initial 2-D position estimate and the final position estimates of the compared filters.

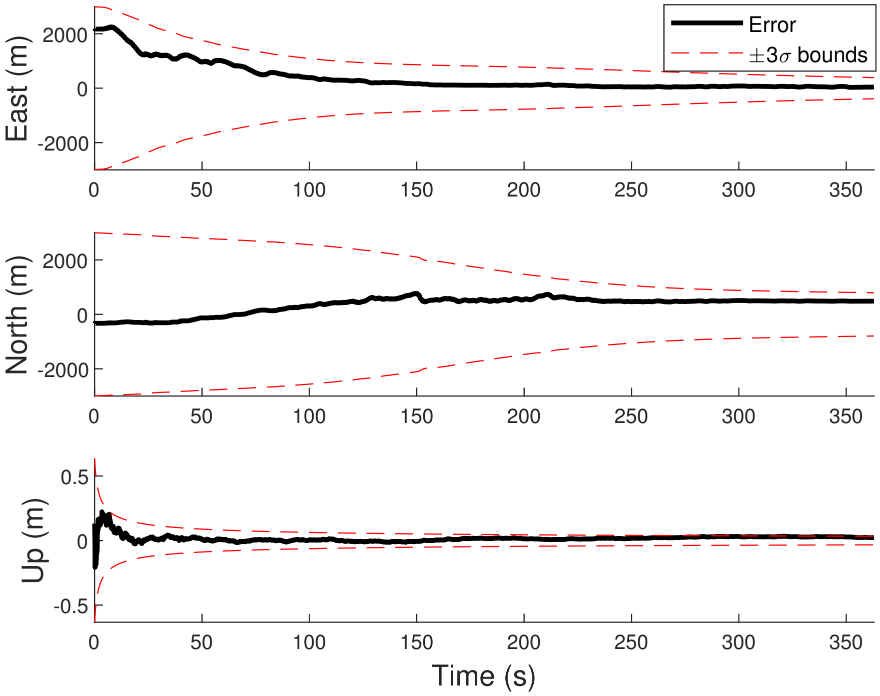


Figure 4.7: EKF position plots for stationary localization using ML generated ephemeris.

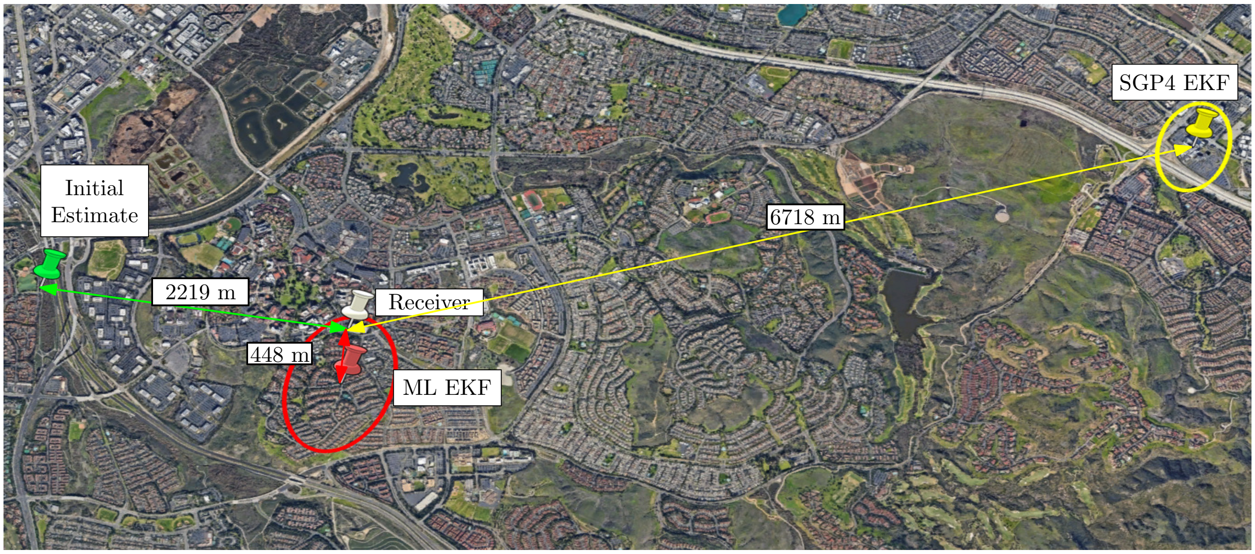


Figure 4.8: Experimental results showing the initial and final 2-D stationary receiver localization errors using (i) SGP4-propagated ephemerides and (ii) ML-propagated ephemerides. Map data: Google Earth.

4.4 Conclusion

This chapter proposed a hybrid numerical-machine learning approach for LEO satellite orbit prediction where the receiver has no prior knowledge on the satellite's position except for publicly available TLE files. A base station tracking satellites along with the chosen suitable ML model could provide receivers with corrected satellite ephemeris for use in localization. The feasibility and accuracy of this approach were showcased experimentally and compared with localization using SGP4 ephemeris (initialized from TLE). The proposed framework reduced the initial receiver error from 2.22 km to 448 m while the SGP4-ephemeris localization yielded a final error of 6,718 m.

Chapter 5

Fusion with Tightly Coupled IMU

5.1 Experiment Overview

A natural extension for integrating the proposed model into a navigation framework is coupling it with an IMU for navigating a ground vehicle. However, to do that, we require observables from multiple satellites for a good navigation solution. As a demonstration, we conduct an experiment where the proposed three stage framework is applied on two Orbcomm satellites, and then on the second time the satellites are observed, their measurements aid a moving ground vehicle in navigating. In this experiment, signals from two Orbcomm satellites, Orbcomm FM113 (NORAD ID 41185) and Orbcomm FM114 (NORAD ID 41179), were collected, and using a specialized software defined radio, carrier-phase measurements were opportunistically extracted. Furthermore, the satellite's downlink signals, which include the satellite's true ephemeris generated by on-board GPS receivers, are decoded for comparison and used as ground truth. Finally, Ionospheric and Tropospheric delays are corrected for in the carrier-phase measurements. The tight coupling of IMU and LEO follows the steps thoroughly described in [84].

The measurements opportunistically extracted from the LEO satellite's signals during the first pass are used to opportunistically track the satellites and refine their ephemeris. Then, this refined ephemeris is used as an input to the supervised training of the hybrid numerical-ML propagator. Each satellite gets a different trained model, and this propagator is subsequently employed to propagate the satellites' ephemeris for around 100 minutes, which corresponds to its orbital period, until the satellites are visible again over Irvine. During the second pass, the measurements extracted from the satellites' signals are tightly coupled with IMU measurements and are used to opportunistically localize a moving vehicle while using the hybrid numerical-ML propagated ephemeris. The localization performance is compared to that obtained using the IMU measurements only.

A suitable time period was chosen where two satellites were available at the same time twice. The satellites are tracked at the first pass, where a stationary receiver with known location receives signals from the satellites and corrects their ephemerides. Then, using the same machine learning approach, the ephemerides are propagated and ready to be used on the moving vehicle.

5.2 Experimental Setup

First, since the second generation active Orbcomm satellites are much fewer than other LEO constellations (12 satellites vs 3000+ satellites eg. Starlink), a suitable time period must be chosen to maximize the availability of multiple satellites at the same time. An algorithm is devised to automate the process:

- All online Orbcomm satellite ephemerides are propagated for multiple days.
- Elevation angles are calculated with respect to a stationary receiver present where the stationary tracking for the first pass occurs.

- Availability of the satellites is calculated based on a 25° elevation mask.
- Windows of time where more than one satellite is available are kept.
- Due to the periodic nature of the satellites' orbits, satellites are available together in a periodic nature. Therefore, there must be multiple windows containing the same satellites.
- Windows with a unique set of satellites, are grouped together. This creates multiple opportunities for conducting the experiments.
- For each group, pairs of windows are created.
- For each group, if any window has a duration less than 2 minutes, it is removed.
- For each group, if the first window in each pair has a satellite with a maximum elevation angle less than 25°, it is removed.
- For each group, the maximum elevations of satellites in the second window during the period of overlap between satellites is recorded. If at least one of of these elevations is less than 35°, the pair is removed.
- For each group, the pairs are filtered by allowing a maximum duration of 5 hours between the first and second window.
- Now we have satisfied multiple criteria:
    - All pairs of windows include at least two satellites.
    - All satellite maximum elevation angles are suitable for both tracking and navigation.
    - All satellite elevation angles are desirable for when the satellites are available at the same time (which is when navigation will occur).

  - The duration between first and second passes is not too long, allowing for more accurately propagated ephemerides to be used during navigation. This is because the accuracy of the propagated ephemeris deteriorates after propagating for a long period of time.
  - The duration of time for tracking and navigating is long enough to allow for convergence or at least conceiving more accurate solutions.

- Since the satellites are all almost at equal altitudes, the elevation angle is correlated with the duration the satellites are available. This means a satellite with a larger elevation angle is available for a longer duration. The final step is to sort the found candidate pairs by overlap duration and ranking them from most overlap duration to least. This information along with the skyplots for the candidate pairs of time windows are created, and the final selection is done manually.

Finally, Figure 5.1 and Figure 5.2 depict the elevation angles and skyplot for the chosen pair respectively. The two pairs are separated by 85 minutes where the machine learning model will propagate the satellites' ephemerides through.

5.3 Stationary Tracking

5.3.1 Experimental Setup

First, we setup the same Orbcomm antenna and USRPs for signal sampling as in Chapters 3 and 4. A very high frequency (VHF) quadrifilar helix antenna was connected to an Ettus E312 Universal Software Radio Peripheral (USRP) to sample Orbcomm LEO satellites' signals at 137-138 MHz at a sampling rate of 2.4 MSps. The USRP's oscillator was disciplined by an external, freely-running NI CDA-2990 OctoClock. The recording was repeated twice

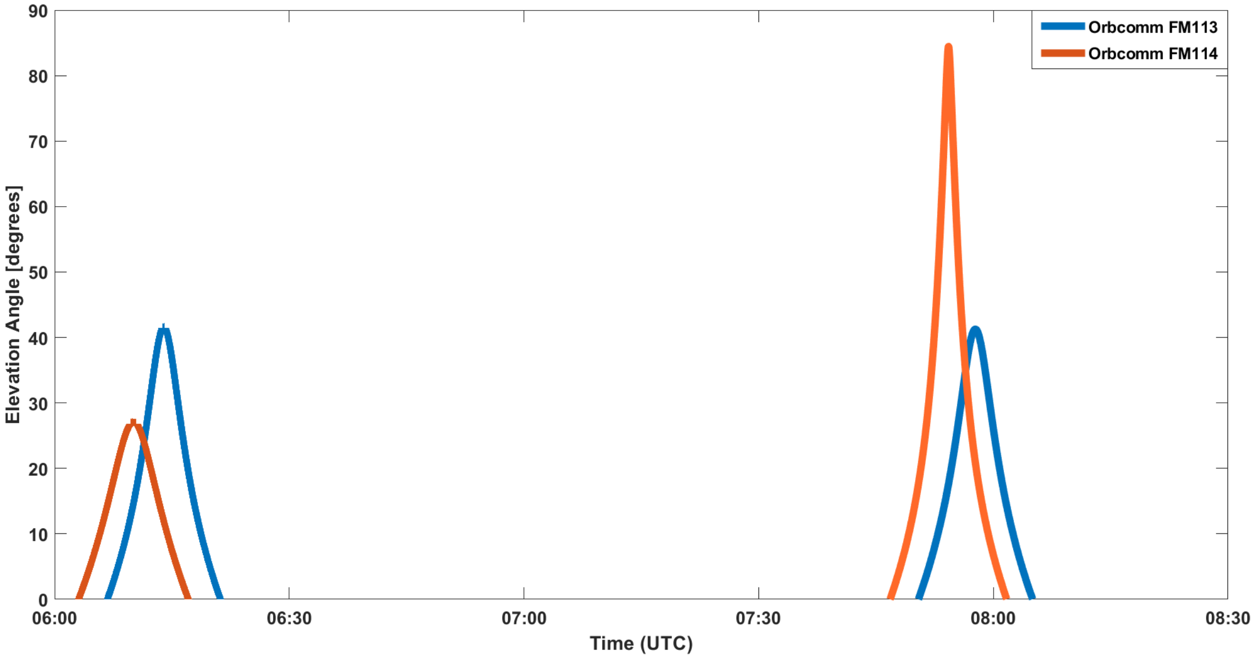


Figure 5.1: Satellite elevation angles for the first and second pass.

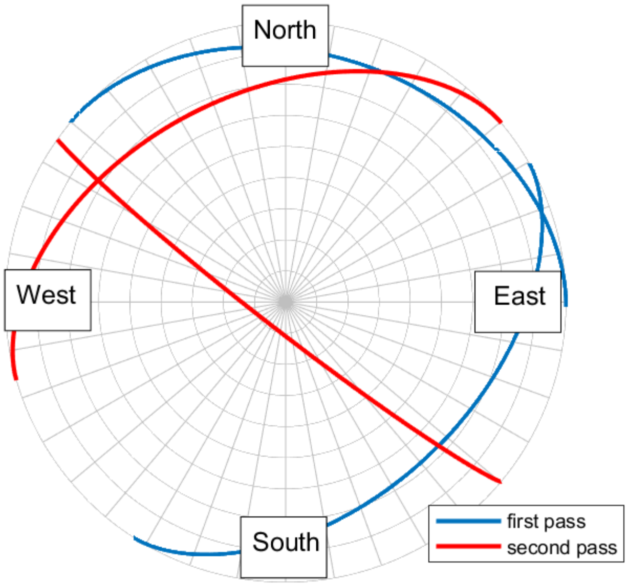


Figure 5.2: Skyplot for the first and second pass.

to record two consecutive passes of Orbcomm FM113 and Orbcomm FM114 LEO satellite over Irvine, California. Since Orbcomm satellites are sparse and do not usually appear in the same elevation mask, the chosen Orbcomm satellites satisfy multiple criteria including maximum elevation, duration for which the satellites are visible, and duration for which both satellites are visible at the same time.

5.3.2 Results

First, a base station with known position starts opportunistically extracting carrier-phase measurements to the satellites. Figure 5.3 plots the transmitted position ephemerides for both tracked satellites.

The sampled observables are Doppler measurements. Each satellite transmits on two different frequencies and thus each satellite has two sets of measurements. Figure 5.4 plots the sampled Doppler measurements for each satellite along with the simulated Doppler measurements using SGP4-propagated ephemeris initialized from TLE. This ephemeris is compared to the transmitted ephemeris and shifted in time to be as close as possible.

The Doppler is noticeably noisy and there are multiple visible outliers. To fix this, a simple outlier removal method is employed. First, the recorded Doppler measurements and the simulated Doppler measurements are subtracted to get an error metric (in Hz). Figure 5.5 shows the Doppler error for each of the satellites.

It is clear that the error oscillates around a certain path, and there are outliers clearly present. To mitigate this, we first fit a line onto the path which will be considered the mean around which noise is additive. Figure 5.6 shows the line fit for each of the error residuals.

Next, each residual error datapoint is subtracted from the fit line. All error residuals that are more than 5 Hz in each direction are removed. The measurements removed are not used at any point during tracked later. Figure 5.7 depicts the remaining residual errors after outlier removal.

Finally, Figure 5.8 plots the corrected Doppler measurements after removing the outliers.

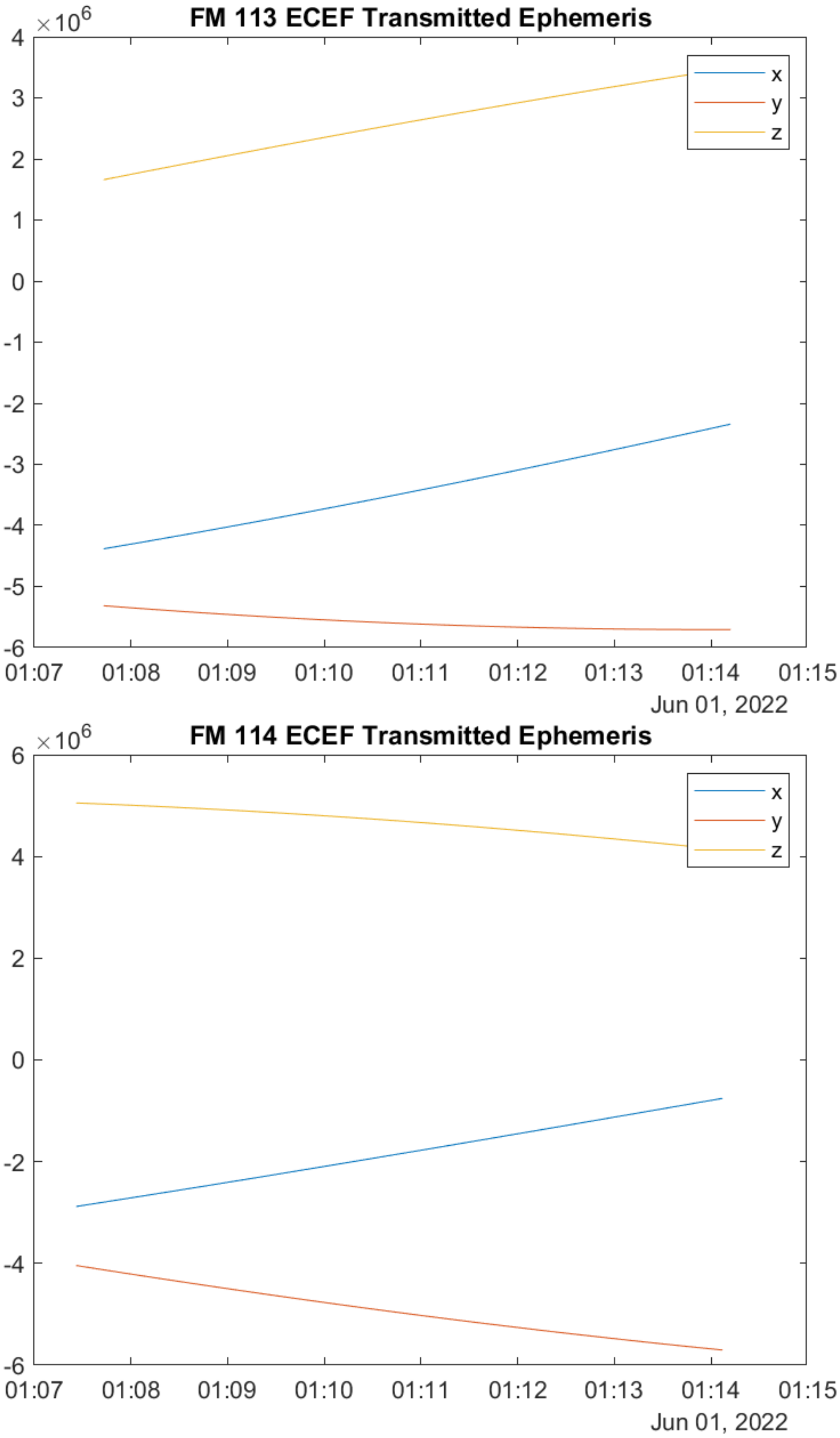
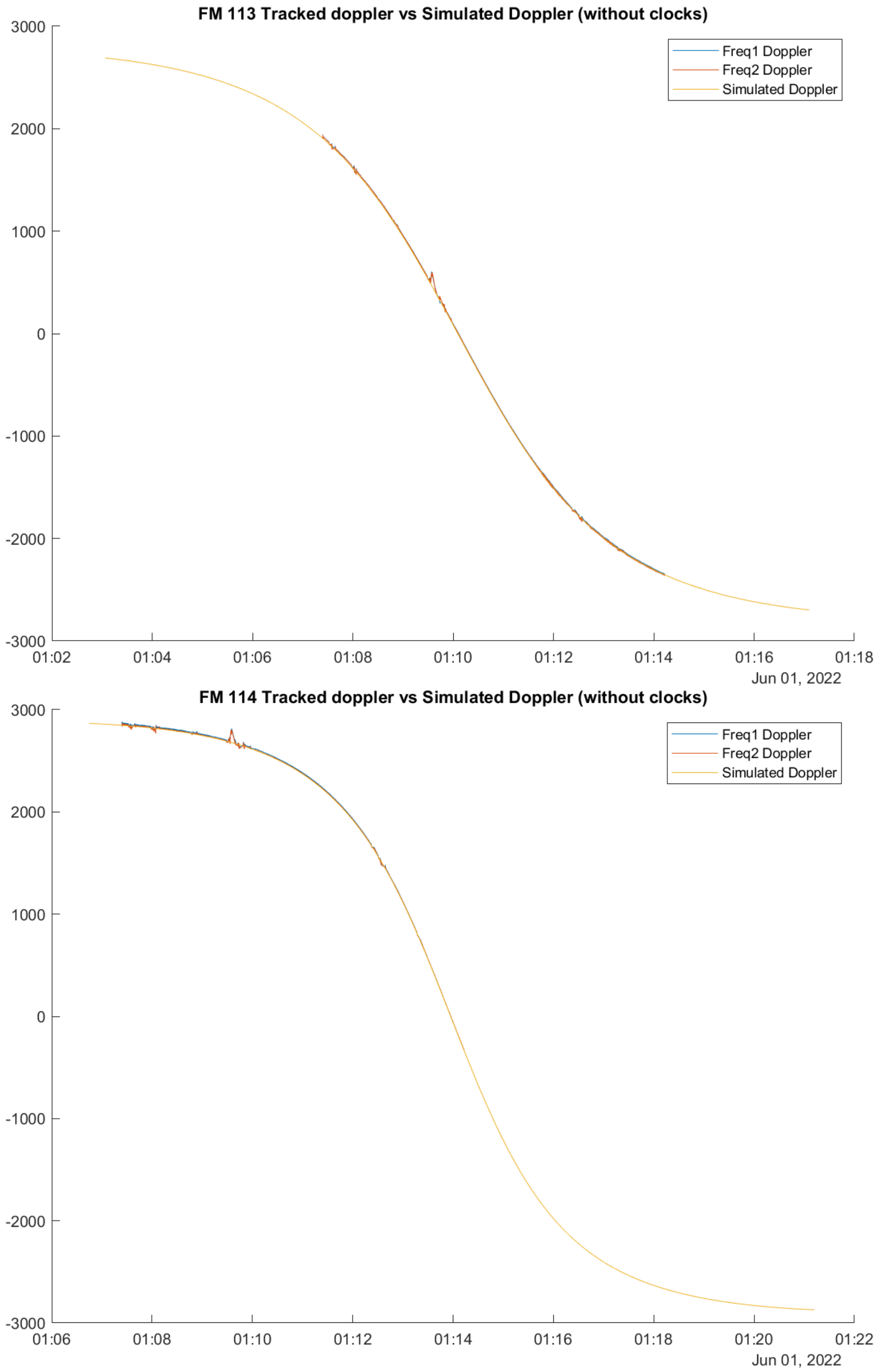


Figure 5.3: Transmitted ephemerides from Orbcomm Satellites FM113 and FM114.


FM 113 Tracked doppler vs Simulated Doppler (without clocks)

FM 114 Tracked doppler vs Simulated Doppler (without clocks)

Figure 5.4: Recorded Doppler measurements for each satellite.

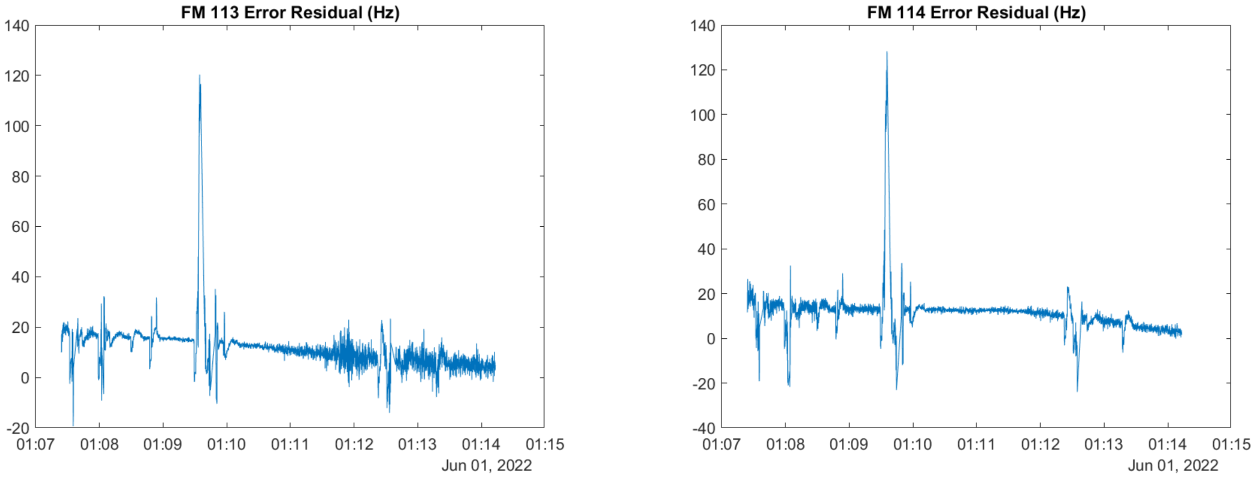


Figure 5.5: Doppler error for each satellite.

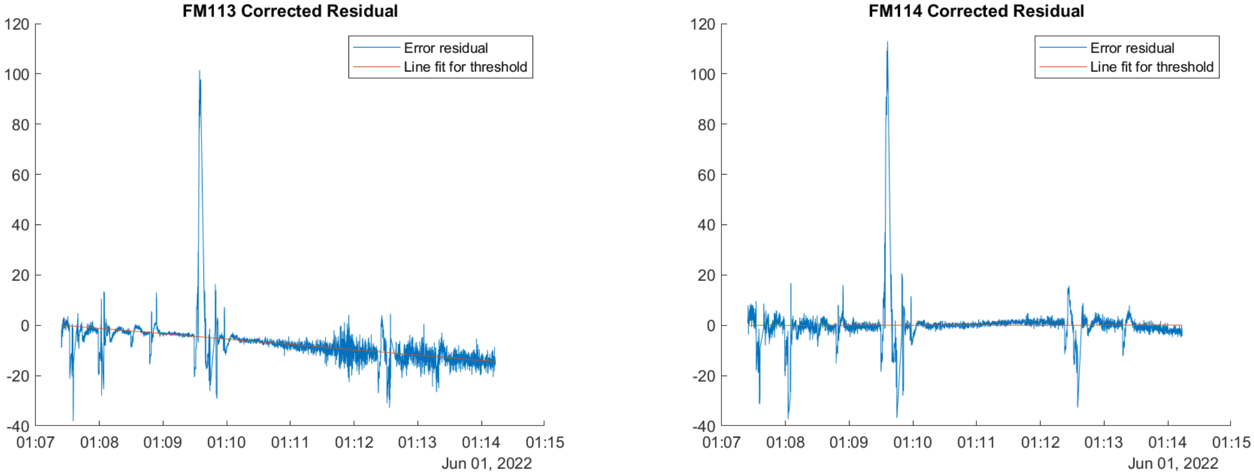


Figure 5.6: Line fit on error residual for each satellites.

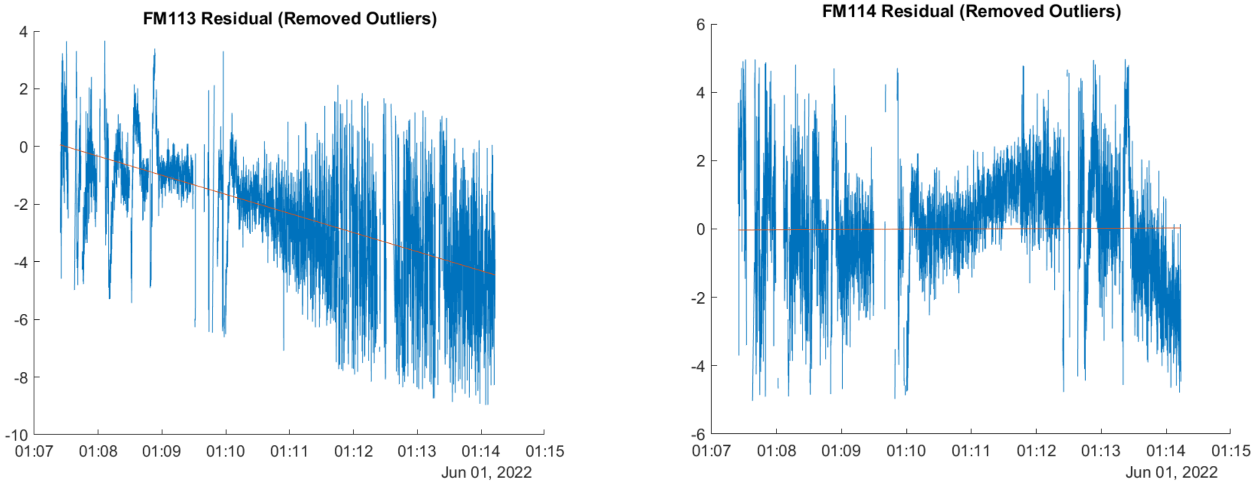


Figure 5.7: Error Residual after outlier removal for each satellite.

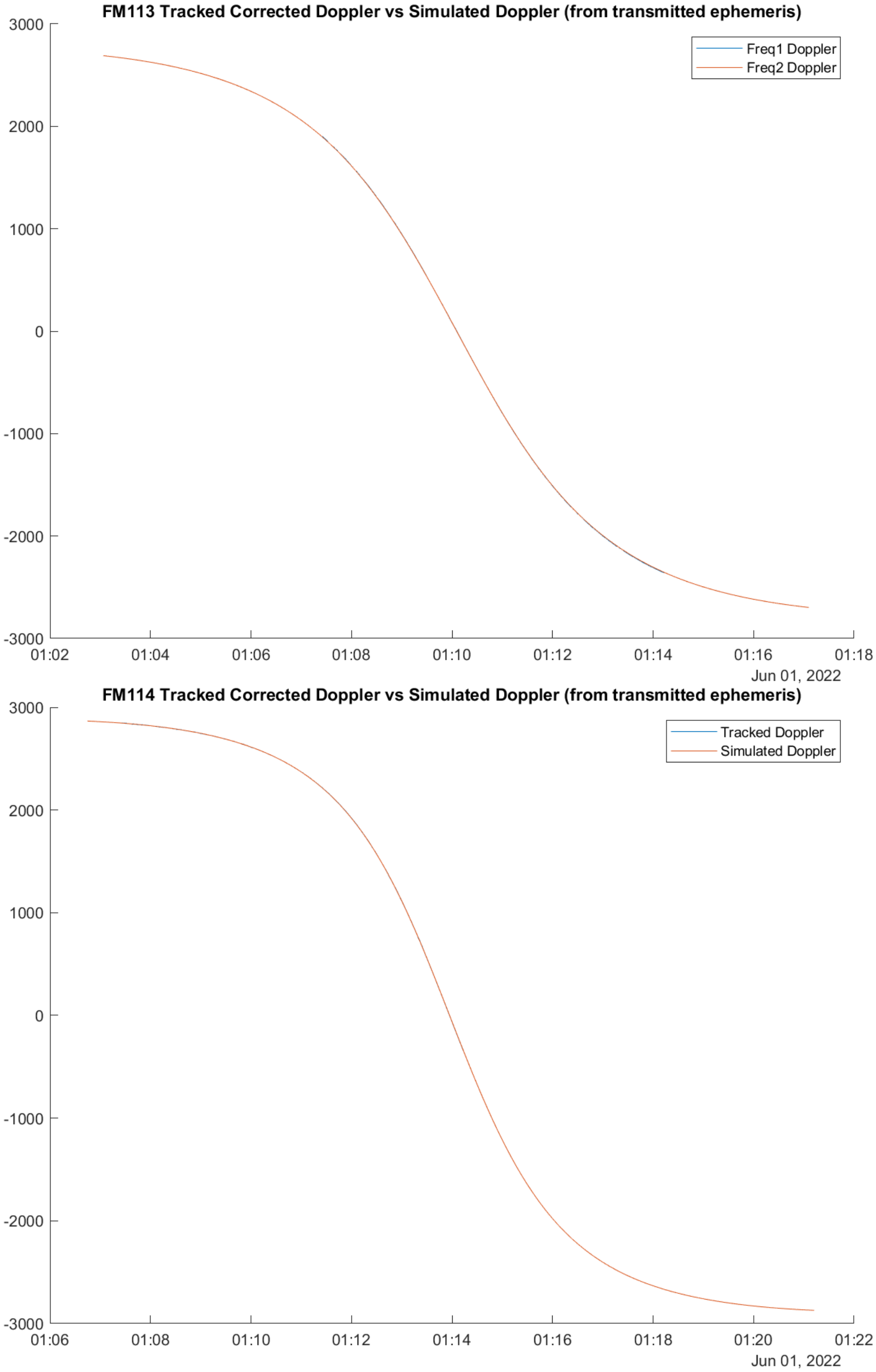


Figure 5.8: Corrected Doppler measurements for each satellite.

The framework described in 4.2.1 is utilized in tracking the satellites and refining their ephemeris. The same framework is utilized as the one in 4.3.2. Orbcomm FM113 was tracked for 389 seconds, 365 seconds of which were usable after removing outliers and measurements that were too inaccurate due to errors at the beginning and end of tracking. Orbcomm FM114 was tracked for 190 seconds. The magnitude of error for Orbcomm FM113 was reduced from 3,091m initial error to 176m final error with reference to the transmitted ephemeris. Figure 5.9 shows the EKF tracking plots in the body frame for the position and velocity of Orbcomm FM113.

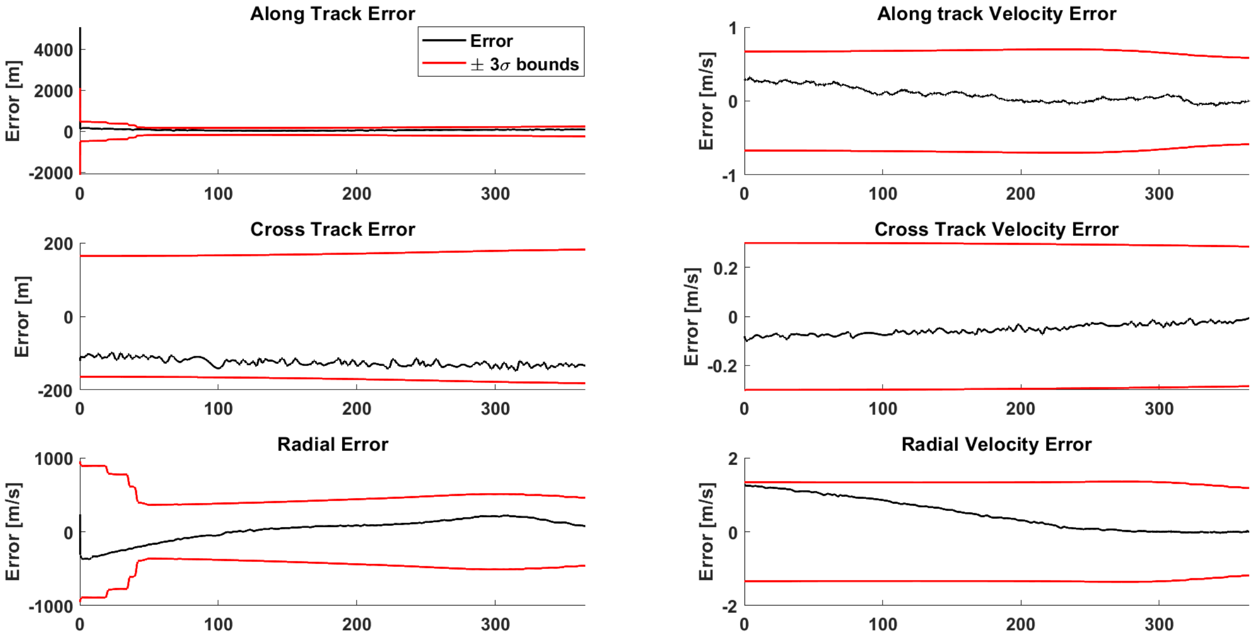


Figure 5.9: Tracking EKF plots for Orbcomm FM113.

Figure 5.10 shows the error magnitude when tracking the satellite's position.

Finally, EKF smoothing is applied allowing us to create target ground truth ephemeris values for training the ML model. The ML model propagates both satellite ephemerides and the generated ephemerides are utilized in navigation as shown in the next section.

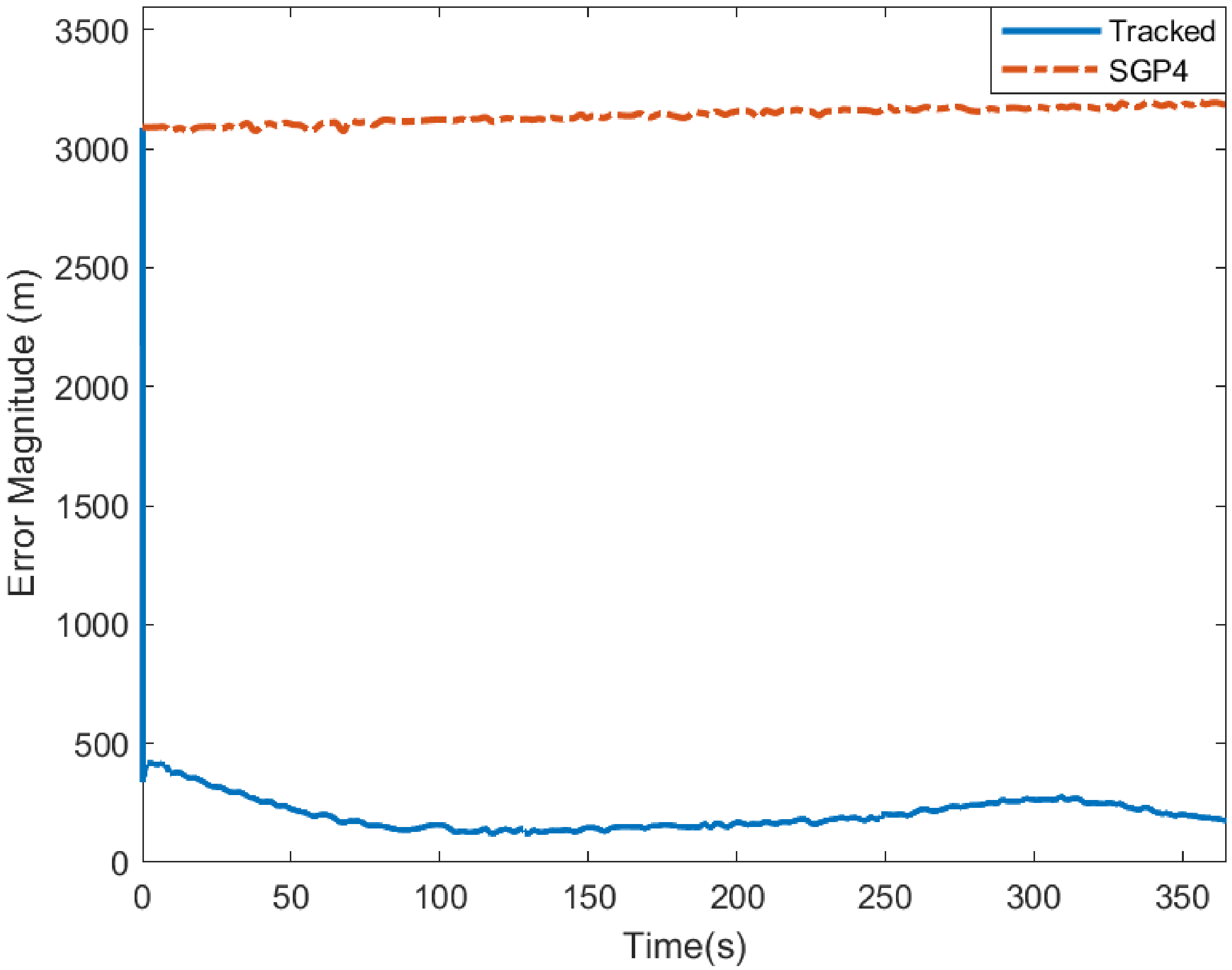


Figure 5.10: Tracking Error Magnitude plots for Orbcomm FM113.

5.4 Tightly Coupled IMU-LEO Navigation on Ground Vehicle

5.4.1 Localization Framework

Figure 5.11 shows the navigation framework at work. The role of the INS in the context of an EKF-based aided INS is to take the sequence of sampled IMU measurements (accelerometer and gyroscope), extract gyroscope and accelerometer biases, and perform successive integrations to propagate an estimate of an IMU state vector between aiding measurement updates.

It is common to directly estimate the orientation, position, and velocity of the IMU in an Earth-centered Earth-fixed (ECEF) frame $\{e\}$ instead of the body frame $\{b\}$, since aiding sources (e.g., GPS satellites and SOP emitters) are typically represented in $\{e\}$. An EKF is developed to estimate the IMU state vector $x_{imu} \in \Re^{16}$ given by:

$$\boldsymbol{x}_{imu} = \left[{}^{b}_{e}\bar{\boldsymbol{q}}, {}^{e}\boldsymbol{r}_{b}^{\mathsf{T}}, {}^{e}\dot{\boldsymbol{r}}_{b}^{\mathsf{T}}, \boldsymbol{b}_{gyr}^{\mathsf{T}}, \boldsymbol{b}_{acc}^{\mathsf{T}}\right]^{\mathsf{T}} \tag{5.1}$$

where ${}^{b}_{e}\bar{\boldsymbol{q}}$ represents the IMU's orientation (i.e rotation from frame $\{e\}$ to $\{b\}$),${}^{e}\boldsymbol{r}_{b}^{\mathsf{T}}$ is the IMU's position, ${}^{e}\dot{\boldsymbol{r}}_{b}^{\mathsf{T}}$ is the IMU's velocity, and $\boldsymbol{b}_{gyr}^{\mathsf{T}}, \boldsymbol{b}_{acc}^{\mathsf{T}}$ represent the IMU's gyroscope and accelerometer biases.

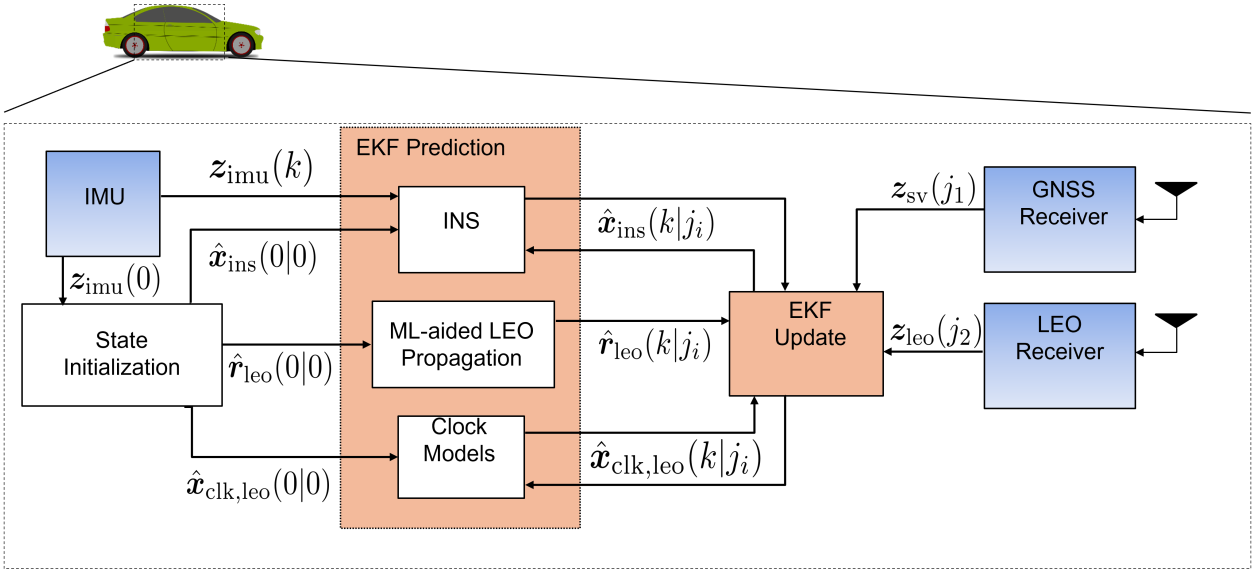

EKF Prediction
IMU
$\boldsymbol{z}_{\text{imu}}(k)$
INS
$\hat{\boldsymbol{x}}_{\text{ins}}(0|0)$
$\boldsymbol{z}_{\text{imu}}(0)$
State Initialization
ML-aided LEO Propagation
$\hat{\boldsymbol{r}}_{\text{leo}}(0|0)$
Clock Models
$\hat{\boldsymbol{x}}_{\text{clk,leo}}(0|0)$
$\hat{\boldsymbol{x}}_{\text{ins}}(k|j_i)$
$\hat{\boldsymbol{r}}_{\text{leo}}(k|j_i)$
$\hat{\boldsymbol{x}}_{\text{clk,leo}}(k|j_i)$
EKF Update
$\boldsymbol{z}_{\text{sv}}(j_1)$
GNSS Receiver
$\boldsymbol{z}_{\text{leo}}(j_2)$
LEO Receiver

Figure 5.11: Navigation Framework.

The EKF prediction propagates the vehicle's states through the IMU, propagates the satellite's states through the ML model, and propagates the clocks as previously described. Finally, the EKF update utilizes GNSS received signals to update and correct the mentioned states. After cutoff, GNSS signals are no longer utilized and LEO signals are used instead.

The INS is loosely coupled with GNSS measurements at first to allow the initial states when GNSS is cut off to have already converged to values close to their true values. This is mainly to allow for correct initial bias estimation. The coupling is loose where the measurements are the position and velocity of the vehicle.

Next, the INS is coupled tightly with LEO measurements once GNSS is cutoff. Appendix 7.6 describes the opportunistic measurement model. Both navigation solutions utilize altimeter measurements. According to the measurement model, the lumped clock terms (the difference between satellite clock bias and receiver clock bias and its derivative) need to be estimated as well. The clock state dynamics are described in Appendix 7.5. The clock states for each satellite is:

$$\boldsymbol{x}_{\text{leo}} = \left[c\Delta\delta t,\, c\Delta\dot{\delta t} \right]^{\mathsf{T}} \tag{5.2}$$

making the final state vector:

$$\boldsymbol{x} = \left[\boldsymbol{x}_{imu}, \boldsymbol{x}_{\text{leo}}^{(1)}, \boldsymbol{x}_{\text{leo}}^{(2)}, \ldots \right] \tag{5.3}$$

The EKF time update, INS state transition and process noise covariance matrices, and measurement update equations are stated in Appendix 7.1,Appendix 7.2, and Appendix 7.3 respectively.

5.4.2 Experimental Setup

Figure 5.12 illustrates the experimental setup for the moving vehicle experiment.

The ground vehicle was equipped with the following hardware and software:

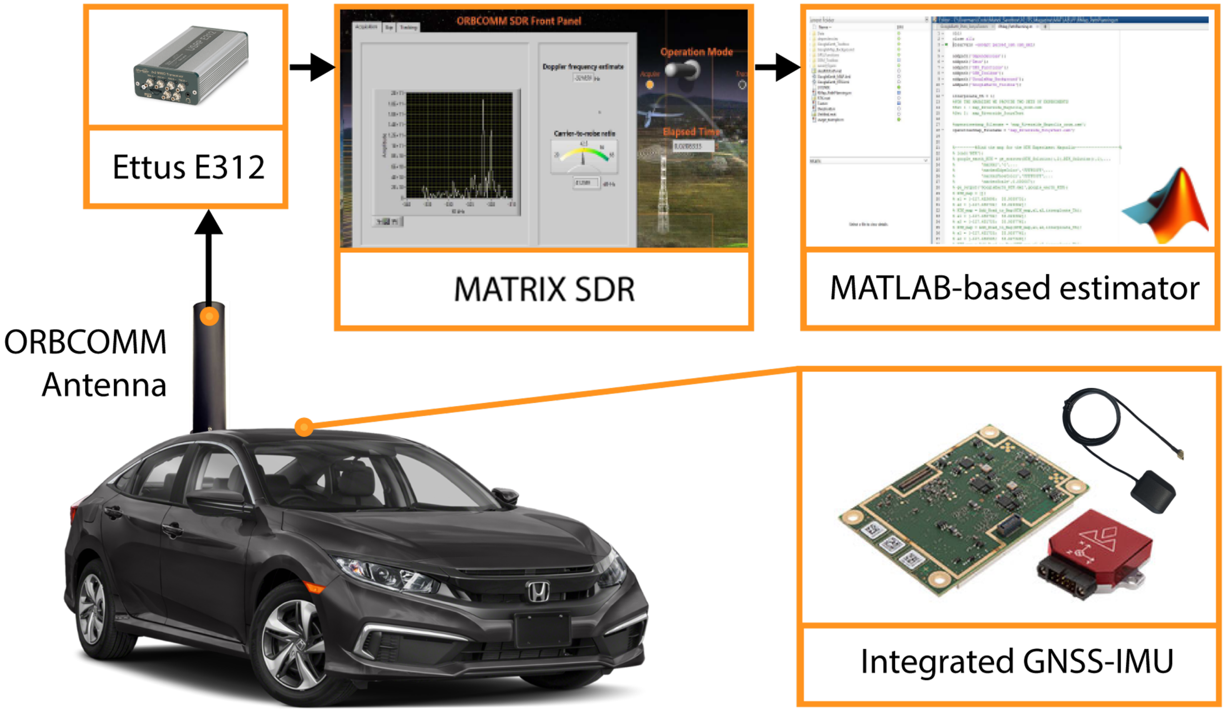


Figure 5.12: Experimental Setup.

- a Septentrio AsteRx-i V integrated GNSS–IMU, which is equipped with a dual-antenna multifrequency GNSS receiver and a Vectornav VN-100 microelectromechanical system IMU; the AsteRx-i V enables access to the raw measurements from this IMU, which was used for the time update of the orientation, position, and velocity, as described in Appendices 7.1,7.2. The carrier phase observables recorded by the Septentrio system were fused by nearby differential GPS base stations to produce the carrier phase-based RTK solution [52]. This RTK solution was used as a ground truth during postprocessing.

- A very high frequency (VHF) quadrifilar helix antenna to capture the Orbcomm satellite signals.

- Ettus E312 Universal Software Radio Peripheral (USRP) to sample Orbcomm LEO satellites' signals at 137-138 MHz at a sampling rate of 2.4 MSps.

- An external, freely-running NI CDA-2990 OctoClock to discipline the USRP's oscillator.

5.4.3 Data Preprocessing

Figure 5.13 shows the simulated and recorded Doppler measurements captured by the moving vehicle. The same pre-processing steps are executed on the recorded measurements, effectively removing outliers.

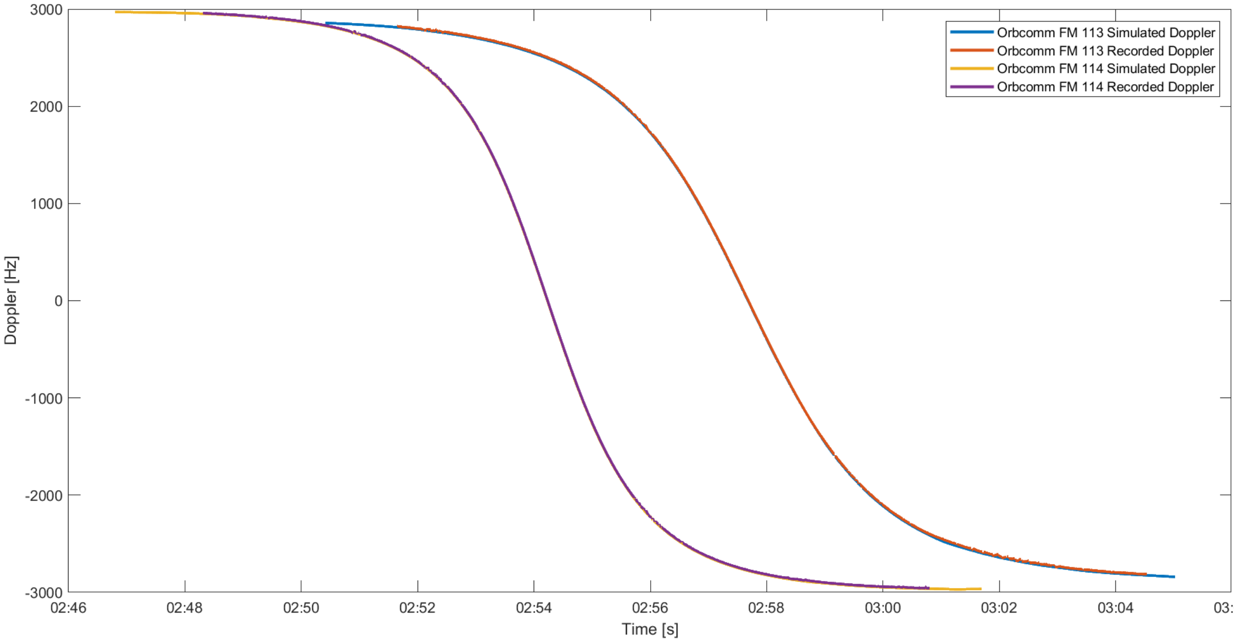


Figure 5.13: Doppler measurements for each satellite.

5.4.4 Experimental Results

The initial section of the navigation solution runs for 40 seconds. This is where the car navigates using IMU measurements coupled loosely with GNSS measurements. This allows the states to converge to estimates close to their true values while their uncertainty decreases. The second section is where both LEO satellites are present. The LEO satellites are captured for a few minutes, but for demonstration purposes, a section of the trajectory is chosen, and the car navigates after GNSS is cutoff for 30 seconds. Figure 5.14 shows the EKF navigation plots for the ground vehicle's position in the ENU (East, North, Up) frame.

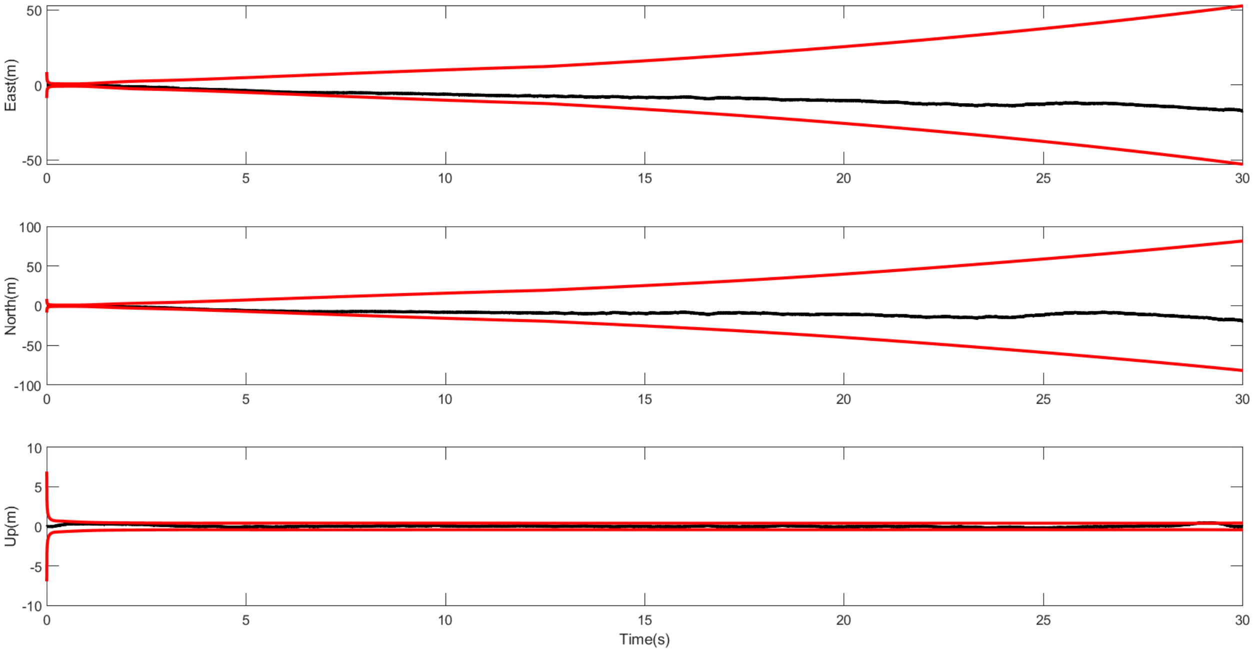


Figure 5.14: EKF position plots for ground vehicle navigation using the ML generated ephemeris.

The ground vehicle was localized for 30 seconds, using the ML-propagated ephemeris as the satellites' ephemerides in the EKF filter. The ground vehicle was also localized utilizing IMU measurements only for 30 seconds. The 2-D magnitude of error when using LEO+IMU increases much slower than that of the IMU-only solution. Figure 5.15 shows the trajectories of both solutions with respect to the true trajectory of the ground vehicle. Figure 5.16 shows the trajectories as seen from Google Earth. It is clear that the LEO+IMU solution outperforms IMU only and provides an adequate navigation solution for short periods of time. This is only the case when the states are initialized close to their true values, when the satellite ephemerides are close to their true values, and when there are multiple satellites present. This demonstration utilizes only two satellites, but it can be extrapolated that when using more satellites, it is expected that the estimates would stay close to their true values for a longer period of time.

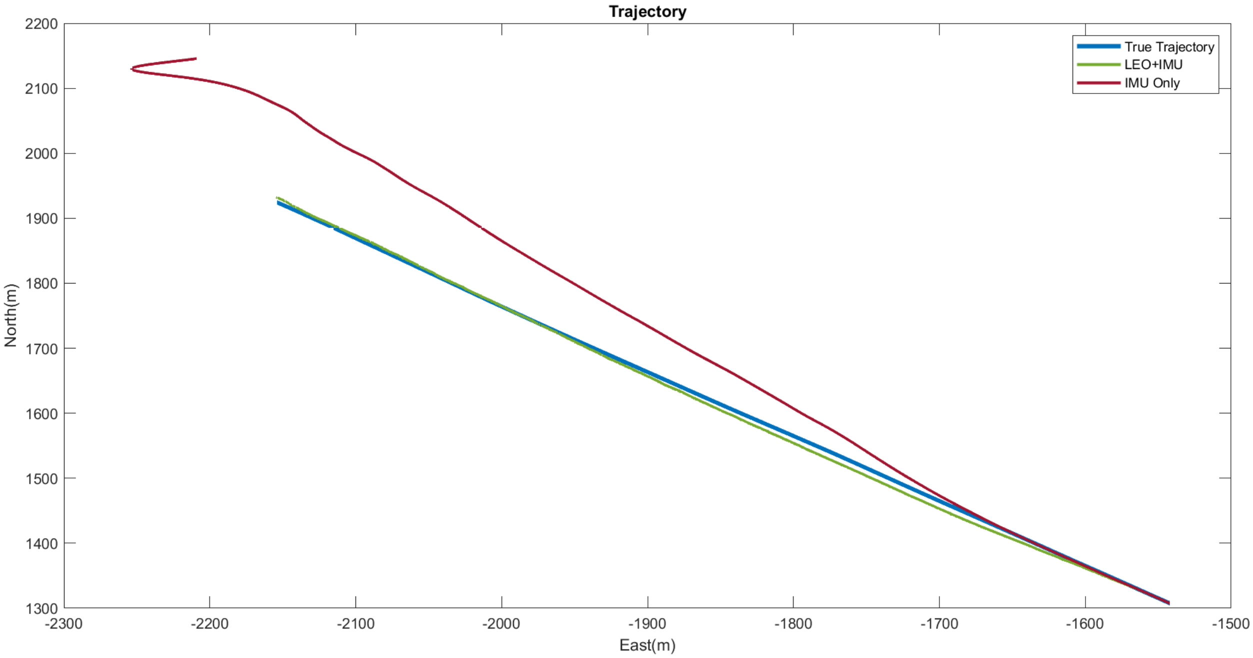


Figure 5.15: Trajectory Comparison between the tightly coupled LEO+IMU navigation solution and the IMU-only navigation solution.

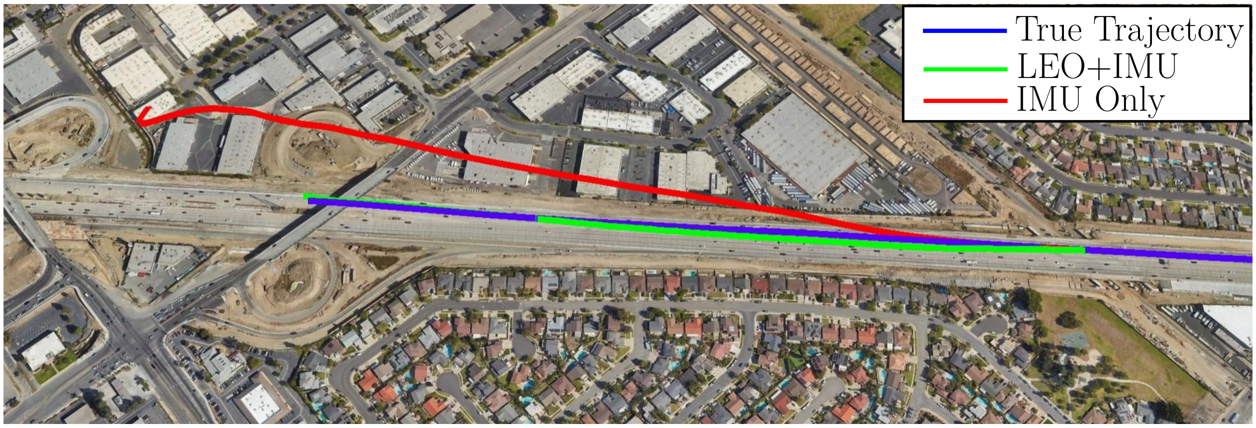


Figure 5.16: Trajectories as seen from Google Earth.

Table summarizes the RMSE and final 2D position error for each of the suggested navigation solutions.

Table 5.1: Comparison of Experimental Results.

Experiment	RMSE(m)	Final Error (m)
IMU Only	105.96	235.5
LEO+IMU	6.00	8.19

5.5 Discussion

It is important to mention that the solution is created after post-processing and many challenges arise for a real-time application. First, the IMU must be initialized and calibrated properly for an accurate navigation solution. In the experiment, GNSS measurements were utilized to initialize values such as accelerometer bias, gyroscope bias, and heading, but in a scenario where GNSS measurements are not available, the IMU must be properly initialized through other means. Furthermore, the satellite ephemeris propagation must not exceed multiple orbits since the deviation of the propagated ephemeris from the satellite's ephemeris will render it unusable for navigation purposes. Ideally, a base station would be tracking satellites and propagating their ephemerides while transmitting propagated ephemerides to a user to use for navigation. Moreover, multiple satellites have to be visible with their ephemerides tracked. This is not a challenge since there is a large amount of satellites from different constellations at any point in time. Finally, the lumped clock states must be initialized close to their true values. Therefore, for a real-time application, the clock states must be correctly initialized either by having GNSS present previously, or through other means (eg. base station).

5.6 Conclusion

This section demonstrated the proposed navigation framework at work. First, the IMU states(orientation, position, velocity, accelerometer bias, and gyroscope bias) are initialized. GNSS measurements are loosely coupled with the IMU measurements by updating the states with position and velocity measurements. The states converge to estimates close to their true values while their uncertainty decreases. GNSS is then cutoff and LEO satellites' measurements are introduced. The state vector is augmented with lumped clock bias and drift terms to account for the LEO satellite measurements. The states are then tightly coupled with measurements from the LEO satellites allowing the error to drift slowly, unlike the IMU-only solution where the error diverges immediately.

Chapter 6

Conclusion

In this thesis, a study of the use of machine learning for satellite orbital determination is conducted. Multiple models for orbit propagation are analyzed, and the model with the best performance is found. The model is then utilized in a STAN framework, experimentally demonstrating its capability of producing satellite ephemeris good enough to allow for desirable navigation solutions. Next, a framework is proposed to collect training data when the target ephemeris data is not available due to satellites not transmitting their ephemeris. Finally, the framework's feasibility is demonstrated experimentally first by localizing a stationary receiver and second by coupling an IMU with LEO observables to navigate a moving ground vehicle.

Chapter 7

Appendix

7.1 IMU Time Update

The tight coupling of IMU and LEO follows the steps thoroughly described in [84]. The time update of $\boldsymbol{x}_{\text{imu}}$ is performed using ECEF strapdown mechanization equations.

7.1.1 Orientation Time Update

The orientation time update is given by

$$ {}^{b}_{e}\hat{\bar{\boldsymbol{q}}}(k+1|j) = {}^{b_{k+1}}_{b_k}\hat{\bar{\boldsymbol{q}}} \otimes {}^{b}_{e}\hat{\bar{\boldsymbol{q}}}(k|j), \tag{7.1} $$

where ${}^{b_{k+1}}_{b_k}\hat{\bar{\boldsymbol{q}}}$ represents an estimate of the rotation quaternion between the IMU's body frame at time k and $k+1$. The quaternion ${}^{b_{k+1}}_{b_k}\hat{\bar{\boldsymbol{q}}}$ is computed by integrating gyroscope rotation rate data $\boldsymbol{\omega}_{\text{imu}}(k)$ and $\boldsymbol{\omega}_{\text{imu}}(k+1)$ using a fourth order Runge-Kutta numerical integration according to

$$\bar{\boldsymbol{\omega}} \triangleq \frac{1}{2}\left[{}^{b}\hat{\boldsymbol{\omega}}(k) + {}^{b}\hat{\boldsymbol{\omega}}(k+1)\right] - {}^{e}_{b}\mathbf{R}^{\mathsf{T}}(k){}^{e}_{i}\boldsymbol{\omega},$$

where $\boldsymbol{\Omega}\left[\,\cdot\,\right] \in \mathbb{R}^{4\times 4}$ is given by

$$\boldsymbol{\Omega}\left[\boldsymbol{a}\right] \triangleq \begin{bmatrix} -\lfloor \boldsymbol{a}\times\rfloor & \boldsymbol{a} \\ -\boldsymbol{a}^{\mathsf{T}} & 0 \end{bmatrix}, \quad \boldsymbol{a} \triangleq \left[a_1, a_2, a_3\right]^{\mathsf{T}},$$

${}^{b}\hat{\boldsymbol{\omega}}(k)$ is the bias-compensated rotation rate measurement, which is computed according to

$${}^{b}\hat{\boldsymbol{\omega}}(k) = \boldsymbol{\omega}_{\text{imu}}(k) - \hat{\boldsymbol{b}}_{\text{gyr}}(k|j), \tag{7.2}$$

and ${}^{e}_{i}\boldsymbol{\omega} \triangleq \left[0,\, 0,\, {}^{e}_{i}\omega\right]^{\mathsf{T}}$ is the rotation rate of the Earth, i.e., the rotation rate of the ECEF frame $\{e\}$ with respect to the ECI frame $\{i\}$. The value of ${}^{e}_{i}\omega$, according to the latest version of the world geodetic system (WGS 84), is ${}^{e}_{i}\omega = 7.292115 \times 10^{-5}$ rad/s.

7.1.2 Position and Velocity Time Update

Integrating IMU specific force data to perform a time update of the position and velocity in an ECEF coordinate frame introduces a centrifugal and Coriolis term due to the rotation rate of the Earth ${}^{e}_{i}\boldsymbol{\omega}$ [85]. Assuming that the variation of the Coriolis force is negligible over the integration interval, the velocity time update is performed according to

$$ {}^{e}\hat{\dot{\boldsymbol{r}}}_b(k+1|j) = {}^{e}\hat{\dot{\boldsymbol{r}}}_b(k|j) + \frac{T}{2}\left[{}^{e}\hat{\boldsymbol{a}}(k) + {}^{e}\hat{\boldsymbol{a}}(k+1)\right] + {}^{e}\boldsymbol{g}(k, {}^{e}\boldsymbol{r}_b(k))T - 2T\lfloor {}^{i}_{e}\boldsymbol{\omega}\times\rfloor {}^{e}\hat{\dot{\boldsymbol{r}}}_b(k|j), \quad (7.3) $$

where ${}^{e}\hat{\boldsymbol{a}}$ and ${}^{b}\boldsymbol{a}$ are the transformed bias-compensated specific force and untransformed bias-compensated specific force, respectively, which are given by

$$ {}^{e}\hat{\boldsymbol{a}}(k) \triangleq \hat{\mathbf{R}}^{\mathsf{T}}(k){}^{b}\hat{\boldsymbol{a}}(k), \quad (7.4) $$

$$ {}^{b}\boldsymbol{a}(k) = \hat{\boldsymbol{a}}_{\text{imu}}(k) - \hat{\boldsymbol{b}}_{\text{acc}}(k|j), \quad (7.5) $$

and $\hat{\mathbf{R}}(k) \triangleq \mathbf{R}\left[{}^{b}_{e}\hat{\bar{\boldsymbol{q}}}(k|j)\right]$.

The position time update is performed according to

$$ {}^{e}\hat{\boldsymbol{r}}_b(k+1|j) = {}^{e}\hat{\boldsymbol{r}}_b(k|j) + \frac{T}{2}\left[{}^{e}\hat{\dot{\boldsymbol{r}}}_b(k+1|j) + {}^{b}\hat{\dot{\boldsymbol{r}}}(k|j)\right] - T^2\lfloor {}^{i}_{e}\boldsymbol{\omega}\times\rfloor {}^{e}\hat{\dot{\boldsymbol{r}}}_b(k|j). \quad (7.6) $$

7.1.3 Accelerometer and Gyroscope Bias Time Update

Noise-corrupted and bias-corrupted IMU measurements can be modeled in discrete-time as

$$ {}^{b}\boldsymbol{\omega}_{\mathrm{imu}}(k) = {}^{b}\,\boldsymbol{\omega}_{i}(k) + \boldsymbol{b}_{gyr}(k) + \boldsymbol{n}_{gyr}(k), \quad k = 1, 2, \ldots \tag{7.7} $$

$$ {}^{b}\boldsymbol{a}_{imu}(k) = {}^{b}_{i}\,\boldsymbol{R}(k)\left[{}^{i}\boldsymbol{a}_{b}(k) - {}^{i}\,\boldsymbol{g}(k, {}^{i}\,\boldsymbol{r}_{b}(k))\right] + \boldsymbol{b}_{acc}(k) + \boldsymbol{n}_{acc}(k) \tag{7.8} $$

where ${}^{b}\boldsymbol{\omega}_{i} \in \Re^{3}$ is the true rotation rate of a coordinate frame $\{b\}$ fixed to the body of the IMU with respect to an inertial frame $\{i\}$, such as the Earth-centered inertial (ECI) frame; $\boldsymbol{b}_{gyr} \in \Re^{3}$ is the gyroscope's three-dimensional (3D) bias; $\boldsymbol{n}_{gyr} \in \Re^{3}$ is a measurement noise vector, which is modeled as a white noise sequence with covariance $\boldsymbol{Q}_{\boldsymbol{n}_{gyr}}$; ${}^{b}_{i}\boldsymbol{R} \in \Re^{3\times 3}$ is the rotation matrix, which rotates the coordinates of a vector expressed in frame $\{i\}$ into frame $\{b\}$; ${}^{i}\boldsymbol{a}_{b} \in \Re^{3}$ is the true acceleration of $\{b\}$ expressed in $\{i\}$; ${}^{i}\boldsymbol{g}_{b} \in \Re^{3}$ is the acceleration due to gravity in the inertial frame, which depends on the position of the IMU ${}^{i}\boldsymbol{r}_{b} \in \Re^{3}$; $\boldsymbol{b}_{acc} \in \Re^{3}$ is the accelerometer's 3D bias; and $\boldsymbol{n}_{acc} \in \Re^{3}$ is a measurement noise vector, which is modeled as a white noise sequence with covariance $\boldsymbol{Q}_{\boldsymbol{n}_{acc}}$. The evolution of the gyroscope and accelerometer biases are modeled as random walks as

$$ \boldsymbol{b}_{gyr}(k+1) = \boldsymbol{b}_{gyr}(k) + \boldsymbol{w}_{gyr}(k) \tag{7.9} $$

$$ \boldsymbol{b}_{acc}(k+1) = \boldsymbol{b}_{acc}(k) + \boldsymbol{w}_{acc}(k) \tag{7.10} $$

The time update of the biases $\boldsymbol{b}_{\text{gyr}}$ and $\boldsymbol{b}_{\text{acc}}$ follow from (7.7) and (7.8), respectively, giving

$$\hat{\boldsymbol{b}}_{\text{gyr}}(k+1|j) = \hat{\boldsymbol{b}}_{\text{gyr}}(k|j),$$

$$\hat{\boldsymbol{b}}_{\text{acc}}(k+1|j) = \hat{\boldsymbol{b}}_{\text{acc}}(k|j).$$

7.2 INS State Transition and Process Noise Covariance Matrices

The calculation of the discrete-time linearized INS state transition matrix $\boldsymbol{\Phi}_{\text{imu}}$ and process noise covariance $\mathbf{Q}_{\text{imu}}$ are performed using strapdown INS equations as described in [86, 85]. The discrete-time linearized INS state transition matrix $\boldsymbol{\Phi}_{\text{imu}}$ is given by

$$\boldsymbol{\Phi}_{\text{imu}} = \begin{bmatrix} \boldsymbol{\Phi}_{\boldsymbol{qq}} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{\Phi}_{\boldsymbol{q}\boldsymbol{b}_{\text{gyr}}} & \mathbf{0}_{3\times3} \\ \boldsymbol{\Phi}_{\boldsymbol{rq}} & \mathbf{I}_{3\times3} & T\mathbf{I}_{3\times3} & \boldsymbol{\Phi}_{\boldsymbol{r}\boldsymbol{b}_{\text{gyr}}} & \boldsymbol{\Phi}_{\boldsymbol{r}\boldsymbol{b}_{\text{acc}}} \\ \boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{q}} & \mathbf{0}_{3\times3} & \boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\dot{\boldsymbol{r}}} & \boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{b}_{\text{gyr}}} & \boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{b}_{\text{acc}}} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} \end{bmatrix},$$

where

$$\boldsymbol{\Phi}_{\boldsymbol{qq}} = \mathbf{I}_{3\times 3} - T\lfloor {}^e_i\boldsymbol{\omega}\times\rfloor, \quad \boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\dot{\boldsymbol{r}}} = \mathbf{I}_{3\times 3} - 2T\lfloor {}^e_i\boldsymbol{\omega}\times\rfloor$$

$$\boldsymbol{\Phi}_{\boldsymbol{q}\boldsymbol{b}_{\text{gyr}}} = -\frac{T}{2}\left[\hat{\mathbf{R}}^{\mathsf{T}}(k+1) + \hat{\mathbf{R}}^{\mathsf{T}}(k)\right],$$

$$\boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{q}} = -\frac{T}{2}\left\lfloor\left[{}^e\hat{\boldsymbol{a}}(k) + {}^e\hat{\boldsymbol{a}}(k+1)\right]\times\right\rfloor, \quad \boldsymbol{\Phi}_{\boldsymbol{rq}} = \frac{T}{2}\boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{q}},$$

$$\boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{b}_{\text{gyr}}} = -\frac{T}{2}\lfloor {}^e\hat{\boldsymbol{a}}(k)\times\rfloor\,\boldsymbol{\Phi}_{\boldsymbol{q}\boldsymbol{b}_{\text{gyr}}}, \quad \boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{b}_{\text{acc}}} = \boldsymbol{\Phi}_{\boldsymbol{q}\boldsymbol{b}_{\text{gyr}}},$$

$$\boldsymbol{\Phi}_{\boldsymbol{r}\boldsymbol{b}_{\text{gyr}}} = \frac{T}{2}\boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{b}_{\text{gyr}}}, \quad \boldsymbol{\Phi}_{\boldsymbol{r}\boldsymbol{b}_{\text{acc}}} = \frac{T}{2}\boldsymbol{\Phi}_{\dot{\boldsymbol{r}}\boldsymbol{b}_{\text{acc}}}.$$

The discrete-time linearized INS process noise covariance $\mathbf{Q}_{\text{imu}}$ is given by

$$\mathbf{Q}_{\text{imu}} = \frac{T}{2}\boldsymbol{\Phi}_{\text{imu}}\mathbf{N}_c\boldsymbol{\Phi}_{\text{imu}}^{\mathsf{T}} + \mathbf{N}_c,$$

where

$$\mathbf{N}_c = \mathrm{diag}\big[\mathbf{S}_{\boldsymbol{n}_{\mathrm{gyr}}},\, \mathbf{0}_{3\times3},\, \mathbf{S}_{\boldsymbol{n}_{\mathrm{acc}}},\, \mathbf{S}_{\boldsymbol{w}_{\mathrm{gyr}}},\, \mathbf{S}_{\boldsymbol{w}_{\mathrm{acc}}}\big]\,,$$

where $\mathbf{S}_{\boldsymbol{n}_{\mathrm{gyr}}} = T\mathbf{Q}_{\boldsymbol{n}_{\mathrm{gyr}}}$ and $\mathbf{S}_{\boldsymbol{n}_{\mathrm{acc}}} = T\mathbf{Q}_{\boldsymbol{n}_{\mathrm{acc}}}$ are the PSD matrices of the gyroscope's and accelerometer's random noise, respectively, and $\mathbf{S}_{\boldsymbol{w}_{\mathrm{gyr}}} = \mathbf{Q}_{\boldsymbol{w}_{\mathrm{gyr}}}/T$ and $\mathbf{S}_{\boldsymbol{w}_{\mathrm{acc}}} = \mathbf{Q}_{\boldsymbol{w}_{\mathrm{acc}}}/T$ are the PSD matrices of the gyroscope's and accelerometer's bias variation, respectively.

7.3 EKF State Measurement Update Equations

The standard EKF equations are modified to deal with the 3-D orientation error correction, which contains one less dimension than the 4-D orientation quaternion estimate. To this end, the state estimate is separated into two parts according to $\hat{\boldsymbol{x}} \triangleq \left[{}^{b}_{e}\hat{\boldsymbol{q}}^{\mathsf{T}}, \hat{\boldsymbol{y}}^{\mathsf{T}}\right]^{\mathsf{T}}$, where ${}^{b}_{e}\hat{\boldsymbol{q}} \in \mathbb{R}^4$ is the orientation quaternion estimate and $\hat{\boldsymbol{y}} \in \mathbb{R}^{14+5M}$ is a vector containing the remaining estimates of $\boldsymbol{x}$. Next, the EKF correction vector $\breve{\boldsymbol{x}}(k+1)$, which is to be applied to the current state prediction $\hat{\boldsymbol{x}}(k+1|j)$ to produce the EKF state measurement update $\hat{\boldsymbol{x}}(k+1|k+1)$, is computed according to

$$\breve{x}(k{+}1) \triangleq \begin{bmatrix} \breve{\boldsymbol{\theta}}(k{+}1) \\ \breve{\boldsymbol{y}}(k{+}1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_{\boldsymbol{\theta}}\mathbf{K}(k{+}1)\boldsymbol{\nu}(k{+}1|j) \\ \boldsymbol{\Lambda}_{\boldsymbol{y}}\mathbf{K}(k{+}1)\boldsymbol{\nu}(k{+}1|j) \end{bmatrix},$$

where $\breve{\boldsymbol{\theta}} \in \mathbb{R}^3$ is the orientation correction, $\breve{\boldsymbol{y}} \in \mathbb{R}^{14+5M}$ is a vector containing the remaining corrections,

$$\boldsymbol{\Lambda}_{\boldsymbol{\theta}} \triangleq \left[\mathbf{I}_{3\times 3}, \mathbf{0}_{3\times(14+5M)}\right],$$

$$\boldsymbol{\Lambda}_{\boldsymbol{y}} \triangleq \left[\mathbf{0}_{(14+5M)\times 3}, \mathbf{I}_{(14+5M)\times(14+5M)}\right],$$

$\boldsymbol{\nu}(k+1|j) \triangleq \boldsymbol{z}(k+1) - \hat{\boldsymbol{z}}(k+1|j)$ is the measurement residual, and $\mathbf{K}$ and $\mathbf{S}$ are defined as

$$\boldsymbol{P}(k+1|k+1) = \boldsymbol{P}(k+1|j) - \boldsymbol{K}(k+1)\boldsymbol{S}(k+1)\boldsymbol{K}^{\mathsf{T}}(k+1) \tag{7.11}$$

$$\boldsymbol{K}(k+1) \triangleq \boldsymbol{P}(k+1|j)\boldsymbol{H}^{\mathsf{T}}(k+1)\boldsymbol{S}^{-1}(k+1) \tag{7.12}$$

$$\boldsymbol{S}(k+1) \triangleq \boldsymbol{H}(k+1)\boldsymbol{P}(k+1)\boldsymbol{H}^{\mathsf{T}}(k+1) + \boldsymbol{R}(k+1) \tag{7.13}$$

Finally, the EKF state measurement update $\boldsymbol{x}(k+1|k+1)$ is computed by applying $\breve{\boldsymbol{\theta}}(k+1)$ to ${}^{b}_{e}\hat{\bar{\boldsymbol{q}}}(k+1|j)$ through

$${}^{b}_{e}\bar{\boldsymbol{q}} = {}^{b}_{e}\hat{\bar{\boldsymbol{q}}} \otimes \tilde{\bar{\boldsymbol{q}}}, \quad \tilde{\bar{\boldsymbol{q}}} = \left[\frac{1}{2}\tilde{\boldsymbol{\theta}}^{\mathsf{T}}, \sqrt{1 - \frac{1}{4}\tilde{\boldsymbol{\theta}}^{\mathsf{T}}\tilde{\boldsymbol{\theta}}}\right]^{\mathsf{T}} \tag{7.14}$$

and applying $\breve{\boldsymbol{y}}(k+1)$ to $\boldsymbol{y}(k+1|j)$ using the standard EKF additive update equation, giving

$$\hat{\boldsymbol{x}}(k+1|k+1) = \begin{bmatrix} {}^{b}_{e}\hat{\bar{\boldsymbol{q}}}(k{+}1|j) \otimes \left[\frac{1}{2}\breve{\boldsymbol{\theta}}^{\mathsf{T}}(k{+}1), \sqrt{1-\frac{1}{4}\breve{\boldsymbol{\theta}}^{\mathsf{T}}(k{+}1)\breve{\boldsymbol{\theta}}(k{+}1)}\right]^{\mathsf{T}} \\ \hat{\boldsymbol{y}}(k{+}1|j) + \breve{\boldsymbol{y}}(k{+}1) \end{bmatrix}. \tag{7.15}$$

7.4 LEO Satellite Dynamics Model

This subsection gives an overview of LEO satellite orbital dynamics, which are used in the navigation filter to estimate the LEO satellites' states in the STAN framework.

A two-body model including the most significant non-zero mean perturbing acceleration is adopted as the LEO satellite orbital dynamics model in the Earth-centered inertial (ECI) reference frame. This model offers a trade-off between accurate open-loop state prediction while maintaining a simple analytical Jacobian for estimation error covariance propagation. The most significant perturbing accelerations for a LEO satellite are due to Earth's non-uniform gravity $\boldsymbol{a}_{\text{grav}}$. The two-body model can be written generally as

$$\ddot{\boldsymbol{r}}_{\text{leo}_l} = \boldsymbol{a}_{\text{grav}_l} + \tilde{\boldsymbol{w}}_{\text{leo}_l}, \qquad \boldsymbol{a}_{\text{grav}_l} = \frac{dU_l}{d\boldsymbol{r}_{\text{leo}_l}}, \tag{7.16}$$

where $\boldsymbol{r}_{\text{leo}_l} \triangleq [x_{\text{leo}_l}, y_{\text{leo}_l}, z_{\text{leo}_l}]^{\mathsf{T}}$ is the position vector of the l-th LEO satellite in the ECI frame, U_l is the non-uniform gravity potential of Earth at the satellite, and $\tilde{\boldsymbol{w}}_{\text{leo}_l}$ is a process noise vector with power spectral density (PSD) $\tilde{\mathbf{Q}}_{\text{leo}_l}$, which attempts to capture the overall acceleration perturbations including the unmodeled non-uniformity of Earth's gravitational field, atmospheric drag, solar radiation pressure, third-body gravitational forces (e.g., gravity of the Moon and Sun), and general relativity [35].

Several models have been developed for U. For a satellite requiring accuracies of a few meters, the JGM-3 model developed by Goddard Space Flight Center is usually sufficient

[87]. Here, the tesseral and sectoral terms of the JGM-3 model are neglected, since they are several orders of magnitude smaller than the zonal terms (denoted $\{J_n\}_{n=2}^{\infty}$). This yields [88]

$$U_l = \frac{\mu}{\|\boldsymbol{r}_{\text{leo}_l}\|}\left[1 - \sum_{n=2}^{N} J_n \frac{R_E^n}{\|\boldsymbol{r}_{\text{leo}_l}\|^n} P_n\left[\sin(\theta_l)\right]\right], \tag{7.17}$$

where μ is Earth's standard gravitational parameter, P_n is a Legendre polynomial with harmonic n, J_n is the n-th zonal coefficient, R_E is the mean radius of the Earth, $\sin(\theta_l) = z_{\text{leo}_l}/\|\boldsymbol{r}_{\text{leo}_l}\|$, and $N = \infty$. Since the acceleration due to the J_2 coefficient is approximately three orders of magnitude greater than the acceleration due to the other zonal coefficients modeling Earth's oblateness, the perturbation due to non-uniform gravity will be approximated by using only the term corresponding to J_2. Taking the partial derivative of (7.17) with respect to the components of $\boldsymbol{r}_{\text{leo}_l}$ with $N \equiv 2$ gives the components of $\boldsymbol{a}_{\text{grav}_l} = \left[\ddot{x}_{\text{grav}_l}, \ddot{y}_{\text{grav}_l}, \ddot{z}_{\text{grav}_l}\right]^{\mathsf{T}}$ as

$$\begin{aligned}
\ddot{x}_{\text{grav}_l} &= -\frac{\mu x_{\text{leo}_l}}{\|\boldsymbol{r}_{\text{leo}_l}\|^3}\left[1 + J_2\frac{3}{2}\left(\frac{R_E}{\|\boldsymbol{r}_{\text{leo}_l}\|}\right)^2\left(1 - 5\frac{z_{\text{leo}_l}^2}{\|\boldsymbol{r}_{\text{leo}_l}\|^2}\right)\right], \\
\ddot{y}_{\text{grav}_l} &= -\frac{\mu y_{\text{leo}_l}}{\|\boldsymbol{r}_{\text{leo}_l}\|^3}\left[1 + J_2\frac{3}{2}\left(\frac{R_E}{\|\boldsymbol{r}_{\text{leo}_l}\|}\right)^2\left(1 - 5\frac{z_{\text{leo}_l}^2}{\|\boldsymbol{r}_{\text{leo}_l}\|^2}\right)\right], \\
\ddot{z}_{\text{grav}_l} &= -\frac{\mu z_{\text{leo}_l}}{\|\boldsymbol{r}_{\text{leo}_l}\|^3}\left[1 + J_2\frac{3}{2}\left(\frac{R_E}{\|\boldsymbol{r}_{\text{leo}_l}\|}\right)^2\left(3 - 5\frac{z_{\text{leo}_l}^2}{\|\boldsymbol{r}_{\text{leo}_l}\|^2}\right)\right].
\end{aligned} \tag{7.18}$$

Further analysis comparing different LEO orbital models can be found in [40, 79].

7.5 Clock Dynamics Model

The receiver's and LEO satellites' clock error state dynamics are assumed to evolve according to [89]

$$\boldsymbol{x}_{\text{clk},i}\left(k+1\right)=\mathbf{F}_{\text{clk}}\,\boldsymbol{x}_{\text{clk},i}(k)+\boldsymbol{w}_{\text{clk},i}(k), \tag{7.19}$$

$$\boldsymbol{x}_{\text{clk},i}\triangleq\left[c\delta t_i,\,c\dot{\delta t}_i\right]^{\mathsf{T}},\quad \mathbf{F}_{\text{clk}}=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix},$$

where $i=\{r,\text{leo}_l\}$, δt_i is the clock bias, $\dot{\delta t}_i$ is the clock drift, c is the speed of light, T is the constant sampling interval, and $\boldsymbol{w}_{\text{clk},i}$ is the process noise, which is modeled as a discrete-time white noise sequence with covariance

$$\mathbf{Q}_{\text{clk},i}=c^2\cdot\begin{bmatrix}S_{\tilde{w}_{\delta t_i}}T+S_{\tilde{w}_{\dot{\delta t}_i}}\frac{T^3}{3} & S_{\tilde{w}_{\dot{\delta t}_i}}\frac{T^2}{2}\\ S_{\tilde{w}_{\dot{\delta t}_i}}\frac{T^2}{2} & S_{\tilde{w}_{\dot{\delta t}_i}}T\end{bmatrix}, \tag{7.20}$$

The terms $S_{\tilde{w}_{\delta t_i}}$ and $S_{\tilde{w}_{\dot{\delta t}_i}}$ are the clock bias and drift process noise PSDs, respectively, which can be related to the power-law coefficients, $\{h_{\alpha_i}\}_{\alpha_i=-2}^{2}$, which have been shown through laboratory experiments to characterize the power spectral density of the fractional frequency deviation of an oscillator from nominal frequency according to $S_{\tilde{w}_{\delta t_i}}\approx\frac{h_{0,i}}{2}$ and $S_{\tilde{w}_{\dot{\delta t}_i}}\approx 2\pi^2h_{-2,i}$ [90]. The receiver's and LEO satellites' process noise covariances $\mathbf{Q}_{\text{clk}_\text{r}}$ and $\mathbf{Q}_{\text{clk}_\text{leo}}$ are calculated from (7.20) using the PSDs associated with the receiver's and LEO satellites' oscillator quality, respectively.

The dynamics of the difference between the receiver's and LEO satellites' clock error states are given by

$$\boldsymbol{\Delta x}_{\text{clk}}(k+1)=\mathbf{F}_{\boldsymbol{\Delta}\text{clk}}\boldsymbol{\Delta x}_{\text{clk}}(k)+\boldsymbol{\Delta w}_{\text{clk}}(k), \tag{7.21}$$

$$\boldsymbol{\Delta x}_{\text{clk}} \triangleq \left[c\Delta\delta t_{\text{leo}_1}, c\Delta\dot{\delta}t_{\text{leo}_1}, \ldots, c\Delta\delta t_{\text{leo}_L}, c\Delta\dot{\delta}t_{\text{leo}_L}\right]^{\mathsf{T}},$$

$$c\Delta\delta t_{\text{leo}_l} \triangleq c\cdot\left[\delta t_{\text{r}} - \delta t_{\text{leo}_l}\right], \quad c\Delta\dot{\delta}t_{\text{leo}_l} \triangleq c\cdot\left[\dot{\delta}t_{\text{r}} - \dot{\delta}t_{\text{leo}_l}\right],$$

where $\mathbf{F}_{\boldsymbol{\Delta}\text{clk}} = \mathbf{I}_{L\times L}\otimes\mathbf{F}_{\text{clk}}$, with $\otimes$ denoting the Kronecker product, and $\boldsymbol{\Delta w}_{\text{clk}}$ is the process noise which has a covariance $\mathbf{Q}_{\boldsymbol{\Delta}\text{clk}}$ that encapsulates the correlation between entries of $\boldsymbol{\Delta x}_{\text{clk}}$ resulting from the common process noise of the receiver clock states. Assuming the LEO satellites to be equipped with identical oscillators, $\mathbf{Q}_{\boldsymbol{\Delta}\text{clk}}$ simplifies to

$$\mathbf{Q}_{\boldsymbol{\Delta}\text{clk}} = \mathbb{1}_{L\times L}\otimes\mathbf{Q}_{\text{clk}_{\text{r}}} + \mathbf{I}_{L\times L}\otimes\mathbf{Q}_{\text{clk}_{\text{leo}}},$$

where $\mathbb{1}_{L\times L}$ is the $L\times L$ matrix with all entries equal to 1 and $\mathbf{I}_{L\times L}$ is the $L\times L$ identity matrix.

7.6 Carrier-Phase Measurement Model

A LEO receiver extracts continuous-time carrier phase measurements from LEO satellites' signals by integrating the Doppler measurement over time [82]. The carrier phase measurement (expressed in meters) made by the receiver on the LEO satellite at time-step k, which represents discrete-time instant $t_k = kT + t_0$ for an initial time t_0, can be modeled in discrete-time as

$$\phi(k) = \left\|\boldsymbol{r}_{\text{r}}(k) - \boldsymbol{r}_{\text{leo}}(k')\right\|_2 + c\left[\delta t_r(k) - \delta t_{\text{leo}}(k'_l)\right] + \lambda N + c\delta t_{\text{iono}}(k) + c\delta t_{\text{tropo}}(k) + v(k), k = 1, 2, \ldots \tag{7.22}$$

where k' represents discrete-time at $t_{k'} = kT + t_0 - \delta t_{\text{TOF}}$, with δt_{TOF} being the true time-of-flight of the signal from the LEO satellite to the receiver, $\boldsymbol{r}_r$ and $\boldsymbol{r}_{\text{leo}}$ are the receiver's and LEO satellite's 3-D position vectors expressed in the same reference frame, λ is the wavelength of the carrier signal transmitted by the LEO satellite, N is the carrier phase ambiguity of the LEO satellite carrier phase measurement, δt_{iono} and δt_{tropo} are the ionospheric and tropospheric delays, respectively, affecting the LEO satellite's signal, and v is the measurement noise, which is modeled as a zero-mean white Gaussian random sequence with variance σ_ϕ^2.

Assuming no cycle slip occurs when the receiver tracks the carrier phase (i.e., the carrier phase ambiguity remains constant), the difference between the receiver and the LEO satellite clock biases and the carrier phase ambiguity are lumped into a single term $c\Delta\delta t(k)$ to simplify the carrier phase measurement model between the receiver and the satellite in (4.1) into

$$\phi(k) = \|\boldsymbol{r}_r(k) - \boldsymbol{r}_{\text{leo}}(k')\|_2 + c\Delta\delta t(k) + c\delta t_{\text{tropo}}(k) - c\delta t_{\text{iono}}(k) + v(k) \tag{7.23}$$

where

$$c\Delta\delta t(k) \triangleq c\left[\delta t_r(k) - \delta t_{\text{sv}}(k')\right] + \lambda N. \tag{7.24}$$

7.7 Ionospheric and Tropospheric Correction Models

Standard models [82] are used to correct for the ionospheric and tropospheric effects on the measurement model (4.1). In general, ionospheric and tropospheric delays can be modeled as the product of the delay at zenith and a mapping function of the LEO SV's elevation angle θ, known as the obliquity factor [82]. As the elevation angle of the SV decreases, the obliquity factor increases due to the fact that signals at low elevation angles propagate longer in the ionosphere and troposphere.

7.7.1 Ionospheric Correction Model

The ionospheric delay at zenith $c\delta t_{\text{iono,z}}$ is typically given by

$$c\delta t_{\text{iono,z}} = \frac{40.3 \times 10^{16}\ \text{TECV}}{f_c^2} \tag{7.25}$$

where f_c is the carrier frequency in Hz, TECV is the total electron content (TEC) in the vertical direction, expressed in TEC units (TECUs). NASA's Jet Propulsion Laboratory (JPL) maintains a real-time global map of the ionosphere's TEC [91]. Additionally, archived TEC maps validated by the International GNSS Service (IGS) using dual-frequency GNSS observations are available at [92].

The ionospheric delay obliquity factor $f_{\text{iono}}(\theta)$ is computed using the thin shell model as [82]

$$f_{\text{iono}}(\theta) = \left[1 - \left(\frac{R_e \cos\theta}{R_e + h_I}\right)^2\right]^{-\frac{1}{2}}, \tag{7.26}$$

where $h_I \approx 350$ km is the mean ionospheric height.

The time-history of the ionospheric delay $c\delta t_{\text{iono}}$ is consequently given by

$$c\delta t_{\text{iono}}(k) = c\delta t_{\text{iono,z}} \cdot f_{\text{iono}}(\theta), \tag{7.27}$$

where the LEO SV's elevation angle θ varies at each time-step k.

7.7.2 Tropospheric Correction Model

The effect of the tropospheric delay on the measurements is split into wet and dry components. The tropospheric wet $c\delta t_{\text{tropo}_w,\text{z}}$ and dry $c\delta t_{\text{tropo}_d,\text{z}}$ delays at zenith are computed using the Hopfield model [82]

$$c\delta t_{\text{tropo}_w,\text{z}} = 0.373\frac{e_0}{T_0^2}\frac{h_w}{5} \qquad\qquad c\delta t_{\text{tropo}_d,\text{z}} = 77.6\times10^{-6}\frac{P_0}{T_0}\frac{h_d}{5}, \tag{7.28}$$

where T_0 is the temperature in Kelvin, P_0 is the total pressure and e_0 is the partial pressure of water vapor in millibars, $h_w = 12$ km, and $h_d \approx 43$ km.

The tropospheric wet $f_{\text{tropo}_w}(\theta)$ and dry $f_{\text{tropo}_d}(\theta)$ obliquity factors are given by

$$f_{\text{tropo}_w}(\theta) = \frac{1}{\sin\theta + \frac{0.00035}{\tan\theta + 0.017}} \qquad\qquad f_{\text{tropo}_d}(\theta) = \frac{1}{\sin\theta + \frac{0.00143}{\tan\theta + 0.0445}} \tag{7.29}$$

The time-history of the tropospheric delay $c\delta t_{\text{tropo}}$ is consequently given by

$$c\delta t_{\text{tropo}}(k) = c\delta t_{\text{tropo}_w,\text{z}}\cdot f_{\text{tropo}_w}(\theta) + c\delta t_{\text{tropo}_d,\text{z}}\cdot f_{\text{tropo}_d}(\theta), \tag{7.30}$$

where the LEO SV's elevation angle θ varies at each time-step k.

Bibliography

[1] Z. Kassas, J. Khalife, A. Abdallah, and C. Lee. I am not afraid of the jammer: Navigating with signals of opportunity in GPS-denied environments. In *Proceedings of ION GNSS Conference*, pages 1566–1585, 2020.

[2] Z. Kassas. Collaborative opportunistic navigation. *IEEE Aerospace and Electronic Systems Magazine*, 28(6):38–41, 2013.

[3] J. Morales, P. Roysdon, and Z. Kassas. Signals of opportunity aided inertial navigation. In *Proceedings of ION GNSS Conference*, pages 1492–1501, September 2016.

[4] Joe Khalife and Zaher M. Kassas. Navigation with cellular CDMA signals—Part ii: Performance analysis and experimental results. *IEEE Transactions on Signal Processing*, 66(8):2204–2218, 2018.

[5] Z. Kassas, J. Khalife, K. Shamaei, and J. Morales. I hear, therefore I know where I am: Compensating for GNSS limitations with cellular signals. *IEEE Signal Processing Magazine*, pages 111–124, September 2017.

[6] Kimia Shamaei and Zaher M Kassas. LTE receiver design and multipath analysis for navigation in urban environments. *Navigation*, 65(4):655–675, 2018.

[7] A. Abdallah, K. Shamaei, and Z. Kassas. Assessing real 5G signals for opportunistic navigation. In *Proceedings of ION GNSS Conference*, pages 2548–2559, 2020.

[8] J. Khalife, M. Neinavaie, and Z. Kassas. Blind Doppler estimation from LEO satellite signals: A case study with real 5G signals. In *Proceedings of ION GNSS Conference*, pages 3046–3054, September 2020.

[9] Z. Kassas, J. Morales, and J. Khalife. New-age satellite-based navigation – STAN: Simultaneous tracking and navigation with LEO satellite signals. *Inside GNSS Magazine*, 14(4):56–65, 2019.

[10] D. Bhattacherjee, W. Aqeel, I. Bozkurt, A. Aguirre, B. Chandrasekaran, P. Godfrey, G. Laughlin, B. Maggs, and A. Singla. Gearing up for the 21st century space race. In *Proceedings of the ACM Workshop on Hot Topics in Networks*, pages 113–119, 2018.

[11] M. Harris. Tech giants race to build orbital internet [news]. *IEEE Spectrum*, 55(6):10–11, 2018.

[12] O. Kodheli, A. Guidotti, and A. Vanelli-Coralli. Integration of satellites in 5G through LEO constellations. In *Proceedings of the IEEE Global Communications Conference*, pages 1–6, 2018.

[13] B. Di, L. Song, Y. Li, and V. Poor. Ultra-dense LEO: Integration of satellite access networks into 5G and beyond. *IEEE Wireless Communications*, 26(2):62–69, 2019.

[14] G. Giambene, S. Kota, and P. Pillai. Satellite-5G integration: A network perspective. *IEEE Network*, 32(5):25–31, 2018.

[15] S. Hackel, O. Montenbruck, P. Steigenberger, U. Balss, C. Gisinger, and M. Eineder. Model improvements and validation of TerraSAR-X precise orbit determination. *Journal of Geodesy*, 91(5):547–562, 2017.

[16] D. King, A. Farrel, and Z. Chen. An evolution of optical network control: From earth to space. In *Proceedings of the IEEE International Conference on Transparent Optical Networks*, pages 1–4, 2020.

[17] D. Cho, W. Choi, M. Kim, J. Kim, E. Sim, and H. Kim. High-resolution image and video CubeSat (HiREV): Development of space technology test platform using a low-cost CubeSat platform. *International Journal of Aerospace Engineering*, 2019.

[18] T. Reid, T. Walter, P. Enge, D. Lawrence, H. Cobb, G. Gutt, M. O'Conner, and D. Whelan. Position, navigation, and timing technologies in the 21st century. Wiley-IEEE, 2021.

[19] Z. Kassas. Position, navigation, and timing technologies in the 21st century. Wiley-IEEE, 2021.

[20] D. Lawrence, H. Cobb, G. Gutt, M. OConnor, T. Reid, T. Walter, and D. Whelan. Navigation from LEO: Current capability and future promise. *GPS World Magazine*, 28(7):42–48, July 2017.

[21] T. Reid, B. Chan, A. Goel, K. Gunning, B. Manning, J. Martin, A. Neish, A. Perkins, and P. Tarantino. Satellite navigation for the age of autonomy. In *Proceedings of IEEE/ION Position, Location and Navigation Symposium*, pages 342–352, 2020.

[22] A. Nardin, F. Dovis, and J. Fraire. Empowering the tracking performance of LEO-based positioning by means of meta-signals. *IEEE Journal of Radio Frequency Identification*, pages 1–1, 2021.

[23] Z. Kassas, M. Neinavaie, J. Khalife, N. Khairallah, J. Haidar-Ahmad, S. Kozhaya, and Z. Shadram. Enter LEO on the GNSS stage – Navigation with Starlink satellites. *Inside GNSS Magazine*, 16(6):42–51, 2021.

[24] R. Landry, A. Nguyen, H. Rasaee, A. Amrhar, X. Fang, and H. Benzerrouk. Iridium Next LEO satellites as an alternative PNT in GNSS denied environments–Part 1. *Inside GNSS Magazine*, 14(3):56–64., May 2019.

[25] F. Farhangian, H. Benzerrouk, and R. Landry. Opportunistic in-flight INS alignment using LEO satellites and a rotatory IMU platform. *Aerospace*, 8(10):280–281, 2021.

[26] J. Khalife, M. Neinavaie, and Z. Kassas. Navigation with differential carrier phase measurements from megaconstellation LEO satellites. In *Proceedings of IEEE/ION Position, Location, and Navigation Symposium*, pages 1393–1404, April 2020.

[27] F. Farhangian and R. Landry. Multi-constellation software-defined receiver for Doppler positioning with LEO satellites. *Sensors*, 20(20):5866–5883, October 2020.

[28] Q. Wei, X. Chen, and Y. Zhan. Exploring implicit pilots for precise estimation of LEO satellite downlink Doppler frequency. *IEEE Communications Letters*, 24(10):2270–2274, 2020.

[29] T. Mortlock and Z. Kassas. Performance analysis of simultaneous tracking and navigation with LEO satellites. In *Proceedings of ION GNSS Conference*, pages 2416–2429, September 2020.

[30] M. Neinavaie, J. Khalife, and Z. Kassas. Acquisition, Doppler tracking, and positioning with Starlink LEO satellites: First results. *IEEE Transactions on Aerospace and Electronic Systems*, 2021. accepted.

[31] R. Cassel, D. Scherer, D. Wilburne, J. Hirschauer, and J. Burke. Impact of improved oscillator stability on LEO-based satellite navigation. In *Proceedings of ION International Technical Meeting*, pages 893–905, January 2022.

[32] Christina Pinell. Receiver architectures for positioning with low earth orbit satellite signals, 2021.

[33] Nadim Khairallah and Zaher M Kassas. An interacting multiple model estimator of leo satellite clocks for improved positioning. In *2022 IEEE 95th Vehicular Technology Conference:(VTC2022-Spring)*, pages 1–5. IEEE, 2022.

[34] B. Schutz, B. Tapley, and G. Born. *Statistical orbit determination.* Elsevier, 2004.

[35] J. Vetter. Fifty years of orbit determination: Development of modern astrodynamics methods. *Johns Hopkins APL Technical Digest*, 27(3):239–252, November 2007.

[36] D. Vallado, P. Crawford, R. Hujsak, and T. Kelso. Revisiting spacetrack report# 3. In *Proceedings of the Astrodynamics Specialist Conference and Exhibit*, page 6753, 2006.

[37] D. Vallado and P. Crawford. SGP4 orbit determination. In *Proceedings of AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pages 6770–6799, August 2008.

[38] North American Aerospace Defense Command (NORAD). Two-line element sets. `http://celestrak.com/NORAD/elements/`.

[39] K. DeMars, R. Bishop, and M. Jah. Entropy-based approach for uncertainty propagation of nonlinear dynamical systems. *Journal of Guidance, Control, and Dynamics*, 36(4):1047–1057, July-August 2013.

[40] X. Tian, G. Chen, E. Blasch, K. Pham, and Y. Bar-Shalom. Comparison of three approximate kinematic models for space object tracking. In *Proceedings of International Conference on Information Fusion*, pages 1005–1012, 2013.

[41] JG Neeion, Paul J Cefola, and Ronald J Proulx. Current development of the Draper semianalytical satellite theory standalone orbit propagator package. *Advances in the Astronautical Sciences*, 97:2037–2052, 1998.

[42] U. Majumder, E. Blasch, and D. Garren. *Deep Learning for Radar and Communications Automatic Target Recognition.* Artech House, Norwood, MA, 2020.

[43] Y. Wang, X. Bai, H. Peng, G. Chen, D. Shen, E. Blasch, and C. Sheaff. Gaussian-binary classification for resident space object maneuver detection. *Acta Astronautica*, 187:438–446, October 2021.

[44] S. Shen, C. Sheaff, M. Guo, E. Blasch, K. Pham, and G. Chen. Three-dimensional convolutional neural network (3D-CNN) for satellite behavior discovery. In *Proceedings of SPIE Sensors and Systems for Space Applications*, volume 11755, pages 1–18, 2021.

[45] S. Sharma and J. Cutler. Robust orbit determination and classification: A learning theoretic approach. *Interplanetary Network Progress Report*, 203:1, 2015.

[46] F. Feng, Y. Zhang, H. Li, Y. Fang, Q. Huang, and X. Tao. A novel space-based orbit determination method based on distribution regression and its sparse solution. *IEEE Access*, 7:133203–133217, 2019.

[47] H. Peng and X. Bai. Limits of machine learning approach on improving orbit prediction accuracy using support vector machine. In *Proceedings of the Advanced Maui Optical and Space Surveillance*, pages 1–22, 2017.

[48] H. Peng and X. Bai. Artificial neural network–based machine learning approach to improve orbit prediction accuracy. *Journal of Spacecraft and Rockets*, 55(5):1248–1260, 2018.

[49] H. Peng and X. Bai. Comparative evaluation of three machine learning algorithms on improving orbit prediction accuracy. *Astrodynamics*, 3(4):325–343, 2019.

[50] N. Salleh, S. Yuhaniz, N. Azmi, and S. Sabri. Enhancing simplified general perturbations-4 model for orbit propagation using deep learning: A review.

[51] T. Mortlock and Z. Kassas. Assessing machine learning for LEO satellite orbit determination in simultaneous tracking and navigation. In *Proceedings of IEEE Aerospace Conference*, pages 1–8, March 2021.

[52] S. Kozhaya, J. Haidar-Ahmad, Z. Kassas, and S. Saab. Comparison of neural network architectures for simultaneous tracking and navigation with LEO satellites. In *Proceedings of ION GNSS Conference*, September 2021. accepted.

[53] J. Sullivan, S. Grimberg, and S. DAmico. Comprehensive survey and assessment of spacecraft relative motion dynamics models. *Journal of Guidance, Control, and Dynamics*, 40(8):1837–1859, 2017.

[54] S. Shuster. *A survey and performance analysis of orbit propagators for LEO, GEO, and highly elliptical orbits.* Utah State University, 2017.

[55] X. Bai and J. Junkins. Modified Chebyshev-Picard iteration methods for orbit propagation. *The Journal of the Astronautical Sciences*, 58(4):583–613, 2011.

[56] J. Aristoff and A. Poore. Implicit Runge-Kutta methods for orbit propagation. In *AIAA/AAS Astrodynamics Specialist Conference*, page 4880, 2012.

[57] Oliver Montenbruck, Eberhard Gill, and Fh Lutze. Satellite orbits: models, methods, and applications. *Appl. Mech. Rev.*, 55(2):B27–B28, 2002.

[58] Analytical Graphics, Inc., Systems Tool Kit (STK). `https://www.agi.com/products/stk`.

[59] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[60] C. Chao, J. Cook, J. Cox, T. Starchville, R. Thompson, and L. Wagner. Independent verification and validation for Analytical Graphics, Inc. of three astrodynamic functions of the Satellite Tool Kit. *Journal of The Aerospace Corporation ATR*, 2000.

[61] M. Refan and A. Dameshghi. GDOP classification and approximation by implementation of time delay neural network method for low-cost GPS receivers. *Iranian Journal of Electrical and Electronic Engineering*, 16(2):192–200, 2020.

[62] V. Peddinti, D. Povey, and S. Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Proceedings of the International Speech Communication Association Conference*, 2015.

[63] X. Yang, H. Sun, X. Sun, M. Yan, Z. Guo, and K. Fu. Position detection and direction prediction for arbitrary-oriented ships via multitask rotation region convolutional neural network. *IEEE Access*, 6:50839–50849, 2018.

[64] H. Kim and T. Bae. Deep learning-based GNSS network-based real-time kinematic improvement for autonomous ground vehicle navigation. *Journal of Sensors*, 2019.

[65] M. Jiménez-Guarneros, P. Gómez-Gil, R. Fonseca-Delgado, M. Ramírez-Cortés, and V. Alarcón-Aquino. Long-term prediction of a sine function using a LSTM neural network. *Journal of Nature-Inspired Design of Hybrid Intelligent Systems*, pages 159–173, 2017.

[66] A. Hatata and M. Eladawy. Prediction of the true harmonic current contribution of nonlinear loads using NARX neural network. *Alexandria Engineering Journal*, 57(3):2018–1518, 2007.

[67] F. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10):2451–2471, 2000.

[68] T. Lin, B. Horne, P. Tino, and C. Giles. Learning long-term dependencies in NARX recurrent neural networks. *IEEE Transactions on Neural Networks*, 7(6):1329–1338, 1996.

[69] H. Siegelmann, B. Horne, and C. Giles. Computational capabilities of recurrent NARX neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 27(2):208–215, 1997.

[70] H. Xie, H. Tang, and Y. Liao. Time series prediction based on NARX neural networks: An advanced approach. In *Proceedings of IEEE International conference on machine learning and cybernetics*, volume 3, pages 1275–1279, 2009.

[71] J. Menezes, P. Maria, and G. Barreto. Long-term time series prediction with the NARX network: An empirical evaluation. *Neurocomputing*, 71(16-18):3335–3343, 2008.

[72] E. Diaconescu. The use of NARX neural networks to predict chaotic time series. *Wseas Transactions on computer research*, 3(3):182–191, 2008.

[73] B. Li, J. Sang, and J. Ning. TLE generation from sparse tracking data and its performance. *Advances in the Astronautical Sciences*, 158:4003–4014, 2016.

[74] M. Zaheer. Kinematic orbit determination of low Earth orbiting satellites, using satellite-to-satellite tracking data and comparison of results with different propagators. Master's thesis, Royal Institute of Technology (KTH), 2014.

[75] D. Racelis and M. Joerger. High-integrity TLE error models for MEO and GEO satellites. In *Proceedings of AIAA SPACE and Astronautics Forum and Exposition*, pages 1–13, September 2018.

[76] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[77] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[78] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. *Advances in neural information processing systems*, 31, 2018.

[79] J. Morales, J. Khalife, U. Santa Cruz, and Z. Kassas. Orbit modeling for simultaneous tracking and navigation using LEO satellite signals. In *Proceedings of ION GNSS Conference*, pages 2090–2099, September 2019.

[80] J. Khalife and Z. Kassas. Receiver design for Doppler positioning with LEO satellites. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5506–5510, May 2019.

[81] Septentrio AsteRx-i V. https://www.septentrio.com/products, 2018.

[82] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2nd edition, 2010.

[83] N. Khairallah and Z. Kassas. Ephemeris closed-loop tracking of LEO satellites with pseudorange and Doppler measurements. In *Proceedings of ION GNSS Conference*, pages 2544–2555, September 2021.

[84] Joshua J Morales and Zaher M Kassas. Tightly coupled inertial navigation system with signals of opportunity aiding. *IEEE Transactions on Aerospace and Electronic Systems*, 57(3):1930–1948, 2021.

[85] P. Groves. *Principles of* GNSS*, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, second edition, 2013.

[86] J. Farrell and M. Barth. *The Global Positioning System and Inertial Navigation*. McGraw-Hill, New York, 1998.

[87] B. Tapley, M. Watkins, C. Ries, W. Davis, R. Eanes, S. Poole, H. Rim, B. Schutz, C. Shum, R. Nerem, F. Lerch, J. Marshall, S. Klosko, N. Pavlis, and R. Williamson. The Joint Gravity Model 3. *Journal of Geophysical Research*, 101(B12):28029–28049, December 1996.

[88] J. Vinti. *Orbital and Celestial Mechanics*. American Institute of Aeronautics and Astronautics, 1998.

[89] R. Brown and P. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, third edition, 2002.

[90] A. Thompson, J. Moran, and G. Swenson. *Interferometry and Synthesis in Radio Astronomy*. John Wiley & Sons, second edition, 2001.

[91] NASA JPL. Ionospheric and atmospheric remote sensing. https://iono.jpl.nasa.gov/.

[92] NASA. NASA's archive of space geodesy data. https://cddis.nasa.gov/archive/gnss/products/ionex/.