Z-CURVE: A TIME DIFFERENCE OF ARRIVALS (TDOA) BASED MODEL FOR GLOBAL NAVIGATION SATELLITE SYSTEMS (GNSS) SIGNALS

A Dissertation

by

GUOYU FU

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Jyh-Charn Liu
Committee Members,	Riccardo Bettati
	Radu Stoleru
	Michael Bishop
Head of Department,	Dilma Da Silva

May 2019

Major Subject: Computer Engineering

ABSTRACT

The high precision timing and positioning messages of GNSS signals are indispensable to numerous modern applications. We can call current level of integration between GNSS and computation as *loosely coupling*, as only x, y, z, t are being passed across two sides. This work aims to explore the next generation of computational GNSS, or *tightly coupling* based on Time Difference of Arrival (TDOA), which may bring benefits in three aspects: cooperative measurements of common events, utilization of deterministic physical behaviors, and low latency of hardware-software integration. For it, this work presents a novel geometric theory called *zero-crossing curve/surface* (z-curve/surface) to characterize signals, time, and orbits of GNSS satellites, in the hope to set a new foundation for more advanced applications of situational awareness. A z-surface is the three dimensional surface where pseudorange measurements to two satellites are equal. A z-curve is the intersection between z-surface and the Earth surface. A z-curve/surface has three main benefits compared to legacy GNSS projections: *deterministic*, *wide area*, *common to multiple receivers*. These properties are helpful for the design of novel digital applications related to GNSS. Starting with characterization of the satellite service areas, this work develops analytical forms (static and dynamic) for the *z-curve* of the GNSS signals, and their parameterization schemes using geometric and numerical tools. Theories proposed in this dissertation are validated with real world experiments.

DEDICATION

This dissertation is dedicated to my parents who devote their lives to educating me as a happy man, and my wife who accompanies me pursuing my dreams.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Steve Liu, and my committee members, Dr. Riccardo Bettati, Dr. Radu Stoleru, and Dr. Michael Bishop, for their guidance and support throughout the course of this research. I would give my deepest appreciation to Dr. Liu for his words of empathy, encouragement, and wisdom. It was those words, together with the enthusiastic recognition from Dr. Bishop and Dr. Zenon Medina-Cetina, that mentally support my independent exploration into this research.

My research work and mental strength were thankfully improved by some inspiring courses and faculty. They are *Distributed Computing* by Dr. Jennifer Welch, *Geometric Modeling* by Dr. Scott Schaefer, *Combinatorics* by Dr. Huafei Yan, and *Pattern Analysis* by Dr. Riccardo Gutierrez-Osuna. Dr. Zhigui Kang from University of Texas at Austin gave me invaluable guidance on GNSS RTK and PPP technologies.

Thanks also go to my friends and colleagues for making my time at Texas A&M University a great experience. Mr. Colton Riedel, Mr. Lingjun Pu, Mr. Donald Beyette, Mr. Zachary Varnadore, and Mr. Tyler Holmes are great partners along the way of my GNSS studies. I would also thank the six undergraduate and graduate students who took my course of *GNSS Receiver Architectures and Algorithms* (GRAA) in Fall 2017. They gave me a pleasant experience of teaching. My fellow labmates Dr. Xing Wang, Mr. Jason Lin, Mr. Benke Qu, and others gave me a lot of joyful memory in this journey. I wish all of them a wonderful future.

During my 6-month internship in Google in 2018, I received warm welcome and patient help from the Android GPS team: Dr. Mohammed Khider, Dr. Frank Van Diggelen, Dr. Wyatt Riley, Dr. Yu-Han Yang, and Mr. Anil Admal. With their help, I adapted from a student to a qualified software engineer in the top notch positioning research group.

Last but not least, thanks to my wife and our parents for their encouragement, patience, and love. Life is always colorful, hopeful, and full of love with them.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Professors Jyh-Charn Liu, Riccardo Bettati, and Radu Stoleru of the Department of Computer Science and Engineering, and Professor Michael Bishop of the Department of Geography.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported by a teaching assistantship from Texas A&M University.

This work was also made possible in part by DOE-National Energy Technology Laboratory under Grant DE-OE0000825. Its contents are solely the responsibility of the author and do not necessarily represent the official views of DOE-National Energy Technology Laboratory.

NOMENCLATURE

Abbreviations

C/A	Coarse acquisition
CDMA	Code Division Multiplexing Addressing
CORS	Continuously Operating Reference Stations
DSP	Digital Signal Processing
ECEF	Earth-Fixed Earth-Centered
ECI	Earth-Centered Inertial
GEO	Geostationary/Geosynchronous earth orbit
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
IGS	International GNSS Service
K-Crossing	The crossing of a k-surface
K-Surface/Curve	K-crossing surface/curve
LEO	Low earth orbit
LOS	Line of sight
MEO	Medium earth orbit
NTRIP	Network Transport of RTCM via Internet Protocol
OOP	Order of pseudoranges
PISI	Parametric-Implicit Surfaces Intersection
PRN	Pseudorandom number

QSIC	Quadratic Surfaces Intersection Curve
RF	Radio frequency
RINEX	Receiver Independent Exchange Format
RK	Runge-Kutta
RTCM	Radio Technical Commission for Maritime Services
SLAM	Simultaneous Localization and Mapping
SSV	Space service volume
SUPL	Secure User Plane Location
TDOA	Time difference of arrivals
TOA	Time of arrival
TOW	Time of week
TSV	Terrestrial service volume
UTC	Universal Time, Coordinated
WGS	World Geodetic System
Z-Crossing	The crossing of a z-surface
Z-Surface/Curve	Zero-crossing surface/curve

Symbols

a_E, b_E	Equatorial and polar radius of the Earth ellipsoid
$\mathbf{d}$	Direction vector of satellite’s radiation
$\mathbb{E}$	The Earth ellipsoid
$\mathbb{R}$, $\mathbb{R}^3$	The set of real numbers, the 3D space of real numbers

o	Origin of the ECEF coordinate, i.e. Earth mass center
$\mathcal{R}, \mathcal{T}$	Rotation matrix, and translation vector of 3D object
S	A satellite
$\mathbb{V}$	Service volume
$\mathbf{x}$	3D ECEF coordinate, i.e. $\mathbf{x} = (x, y, z)^{\boldsymbol{T}}$
λ, φ, ψ	Longitude, geocentric latitude, geodetic latitude
θ	The target point of satellite’s antenna

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER I

INTRODUCTION

I.1 Challenges of GNSS Based Digital Applications

GNSS/GPS has been a mature technology serving as a backbone of modern society, since it was firstly launched in 1970s and fully open to the public by President Clinton who cancelled the Selective Availability of GPS C/A code [31]. GNSS equipped devices are aware of current position and global time, which enables tremendous applications: location based services, geographic surveying, precision agriculture, aviation navigation, autonomous driving and flying, timestamping of network, finance, and smart grid applications. Billions of devices are involved in these and more GNSS based applications [99]. These devices have been improving rapidly in terms of computational intelligence, thanks to the Moore's law and data sharing brought by mobile Internet and data mining. I believe the present integration of GNSS technology and computational power has not yet fully released their potentials. Position and time values (x, y, z, t) are the only information exchanged between the two sides. Even though Assisted GNSS (AGNSS) and Differential GNSS (DGNSS) collect accessory information for GNSS equipment using Internets, for example, the SUPL protocol [100] and the NTRIP protocol [25], their ultimate goal is to quickly and precisely obtain x, y, z, t. We can call this level of integration as *loosely coupling* between GNSS and computing science.

Loosely coupling treats other sides as if a black box, which would overlook at least three potential benefits. *First of all*, details are ignored or filtered out. Digital applications fail to take complete advantage of the highly deterministic behaviors of satellites, and atomic clocks (or the most advanced clock now – hydrogen maser [26]). Instead, these insights in each signal channel are fused into a single solution, which may have been compromised by some bad or spoofed channel(s). GNSS devices do not fully enjoy the digital resources, such as precise maps, sensors, and local environmental specifics. Fortunately, this has been gradually realized in research communities related to precise urban positioning, SLAM, and GNSS-Reflection, in order to assist the GNSS Time-Of-Arrival (TOA) measurements. *Second of all*, GNSS receivers can only do measurement independently, but not cooperatively. Satellite orbiting, signal broadcasting, atmospheric refracting, these are global events common to receivers in respective proximities. Cooperatively measuring these common events enables the collection of correlated information from different views of many receivers, to calibrate models of event factors. It may also serve as a signal to trigger scheduled tasks of remote receivers, or a monitor to find out anomalous receivers that do not sense the common event on the right time. These cooperation applications are not enabled now, mainly because the TOA projection is specific to each individual receiver. The key to enable them is to find a different projection that maps from satellites, clocks, and signals to a group of receivers. I believe Time Difference of Arrivals (TDOA) is a good fit for this purpose, which will be elaborated further in section I.2. *Third of all*, the latency of computing a solution is high (0.5-1 seconds [25]) and inevitable. The computation time is used for waiting all signal channels and smoothing measurements over time, for more precise solutions. A low latency integration of computing units and GNSS processing units would enable more advanced applications: as soon as measuring a GNSS event, the computing unit can immediately react, without waiting for other channels or smoothing. Ideally, the minimal latency between sensing and reaction is the correlation time of CDMA (Code Division Multiplexing Addressing) codes of GNSS, which should be few cycles of modern DSP (Digital Signal Processing) chip.

These challenges are asking for a *tight coupling* scheme of the following features: first, it should operate on the level of signal channel rather than the level of solution; second, it should be a new projection of satellites, signal, and times to a group of receivers. The traditional perspective based on solving a group of TOA measurements is not suitable to tackle these challenges. In addition, each TOA measurement from a satellite to a receiver is not an independent projection, because it contains the receiver's clock bias, which is one of the unknowns and different from receiver to receiver. As such, this dissertation takes a non-TOA perspective (TDOA) to explore possible tightly coupling schemes to mitigate these three drawbacks by the traditional loosely coupling between computation and GNSS.

I.2 TDOA Based GNSS

Unlike TOA that uses a single pseudorange, TDOA measures the difference of pseudoranges from two different satellites. TOA projection is a sphere with receiver-specific pseudorange as radius, while TDOA projection is a hyperboloid with two satellites as foci. The hyperboloid is non-specific to receiver model, because the difference operation cancels out the common term of receiver clock bias in two pseudoranges.

In this dissertation, I call the TDOA surface as *k-surface*, or formally as the set of points that measures the difference of two pseudoranges to be a real number *k*. A special case is called *z-surface*, where two pseudoranges are equal (i.e. difference is zero). There are several benefits of *z-surface* versus individual pseudoranges. *Firstly*, it is independent of receiver models, and thus can be precisely modeled into a function of orbits, atmospheric delays, atomic clocks, and signal characteristics. *Secondly*, it spreads inter-continentally so that receivers in a wide area can observe such a common phenomenon. Remote scheduling, crowd sourcing checking, and other distributed applications can then be developed based on this principle. *Thirdly*, *z-surface*'s behavior is highly deterministic, and even periodic. As satellites orbits cyclically over hours (e.g. periods of GPS and GLONASS satellites are 11 hours 58 minutes and 11 hours 5 minutes [101]), the motion of z-surface is alike whose period should be the common multiple of the periods of two incident satellites. This helps applications in terms of repeated observations and look-up table implementation. *Fourthly*, it converts the real value of pseudorange into a discrete value or event. Instead of pseudoranges, a z-surface separates receivers into its two sides, where in one side a pseudorange is larger than the other and vice versa in the other side. Receivers can know the order of pseudoranges (OOP) by figuring out which side of z-surface they are locating at. They know the change of OOP at the time when the z-surface comes across them, which is termed a *zero-crossing event* or *z-event* in this dissertation. This event can be promptly detected in the tracking loop of receivers.

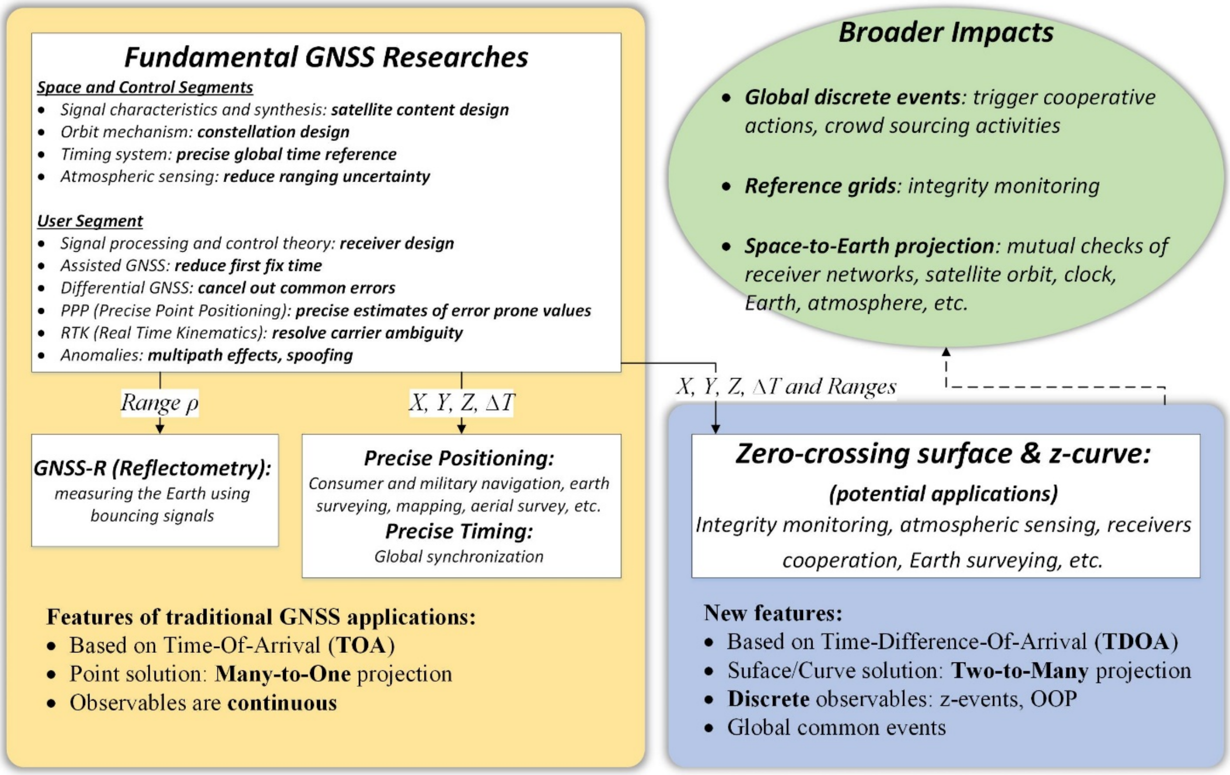

Fundamental GNSS Researches
Space and Control Segments
Signal characteristics and synthesis: satellite content design
Orbit mechanism: constellation design
Timing system: precise global time reference
Atmospheric sensing: reduce ranging uncertainty
User Segment
Signal processing and control theory: receiver design
Assisted GNSS: reduce first fix time
Differential GNSS: cancel out common errors
PPP (Precise Point Positioning): precise estimates of error prone values
RTK (Real Time Kinematics): resolve carrier ambiguity
Anomalies: multipath effects, spoofing
Range ρ
X, Y, Z, ΔT
GNSS-R (Reflectometry): measuring the Earth using bouncing signals
Precise Positioning: Consumer and military navigation, earth surveying, mapping, aerial survey, etc.
Precise Timing: Global synchronization
Features of traditional GNSS applications:
Based on Time-Of-Arrival (TOA)
Point solution: Many-to-One projection
Observables are continuous
Broader Impacts
Global discrete events: trigger cooperative actions, crowd sourcing activities
Reference grids: integrity monitoring
Space-to-Earth projection: mutual checks of receiver networks, satellite orbit, clock, Earth, atmosphere, etc.
X, Y, Z, ΔT and Ranges
Zero-crossing surface & z-curve:
(potential applications)
Integrity monitoring, atmospheric sensing, receivers cooperation, Earth surveying, etc.
New features:
Based on Time-Difference-Of-Arrival (TDOA)
Suface/Curve solution: Two-to-Many projection
Discrete observables: z-events, OOP
Global common events

Figure I-1 Innovation of this dissertation in GNSS research

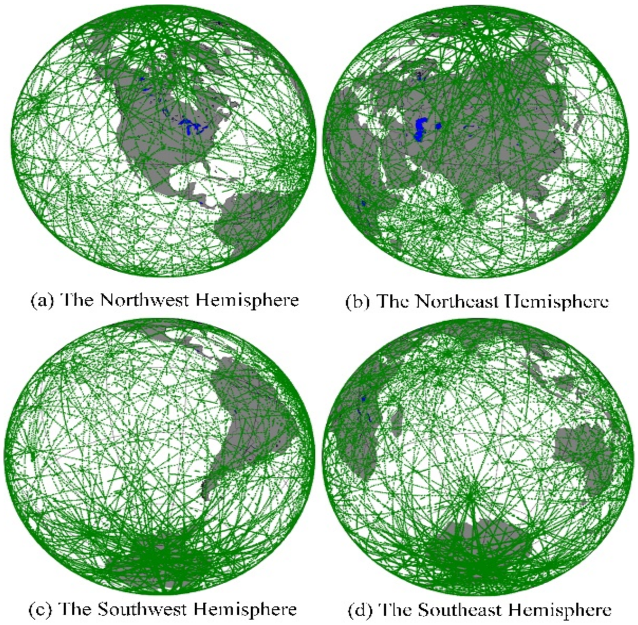
Not many studies on TDOA projection were published. The existing researches only use it to calculate a user's position and time. For instance, the Loran-C, eLoran, and the British Gee system use TDOA, and are considered as backup for GPS [102]. Books discussing TDOA only illustrate it in a planar canvas, without elaborate discussion [25][26][28]. This work aims to formalize the theory of z-surface, including its physical models, covering ranges, dynamic behaviors, and potential applications.

Novelty of this dissertation is summarized in Figure I.1. The proposed z-surface theory takes advantage of well-established fundamental GNSS models such as receiver design, and orbital mechanism. I aim to build fundamental z-surface theory for the exploration of new applications based on TDOA beyond traditional positioning and timing. Potential applications include but not limit to GNSS integrity monitoring, crowd sourcing activities, mutual checks of receiver networks, satellite orbits, clock, Earth surface, and atmosphere models.

I.3 Statement of Research Problems

It is already known that geometric modeling of TDOA can be effectively developed on a hyperbolic framework, but most existing work were based on the planar environment, not for 3D free space [20]. New geometric modeling techniques and their solution systems are needed to characterize TDOA surfaces in 3D space, as well as on the Earth surface. Moreover, even though the orbital mechanism is a well-studied field, it is rarely seen any published works for applying satellite dynamics on the motion of z-surface.

This dissertation aims to develop fundamental models to understand the characteristics and dynamics of z/k-surface, and its intersection with the Earth surface, in the hope to enable development of novel applications based on TDOA of GNSS. I aim to investigate static and dynamic properties of GNSS signals derived from their TDOA measures. The investigation starts with the notion of Order of Pseudoranges (OOP) that aims to use snapshots of pseudoranges to characterize the states of GNSS signals at a position, which will be detailed in Chapter I. Based on this raw model in [1], I then continued to develop a simple point-based model to plot all GPS z-curves for a time snapshot in Figure 1. When one unfreezes the time, the z-surface/curve will move and morph as the satellites fly, and the atmospheric refraction varies, as shown in Figure 2.


(a) The Northwest Hemisphere

(b) The Northeast Hemisphere

(c) The Southwest Hemisphere

(d) The Southeast Hemisphere

Figure I-2 All GPS z-curves on 00:00 May 25, 2015

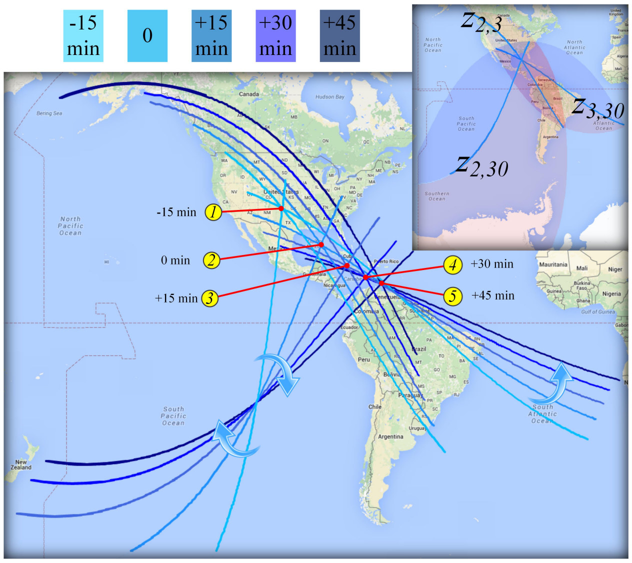


Figure I-3 Moving z-curves of GPS satellites in one hour (15-minute interval)

I was deeply amazed by these unforeseen pictures generated from my simple model. More questions were raised in my mind: what is the bound of each curve at two ends? How “thick” is each curve? Why they have different curvatures? How fast do they move? How long will they exist? How can we compactly represent them and use them in computer world? Answering these questions may help development of advanced applications based on novel properties derived from TDOA measures. On one hand, novel distributed systems may utilize the intersecting k-curves to segment ground area, where in each small region is identified by a certain OOP. These region identities are naturally incorporated with global time and position information, which may be used to design integrity checking, scheduling, communication mechanism for wide-area infrastructures, location-based service [18][19] etc. On the other hand, precise geodesic and aerospace measurement may benefit from the proposed models. By mapping between the precise navigation satellites and a group of ground receivers (instead of individual receivers), a good number of correlated measurements can be obtained to build statistical profiles of observables including crustal movement, glacier flow, microgravity value, precise orbits, etc.

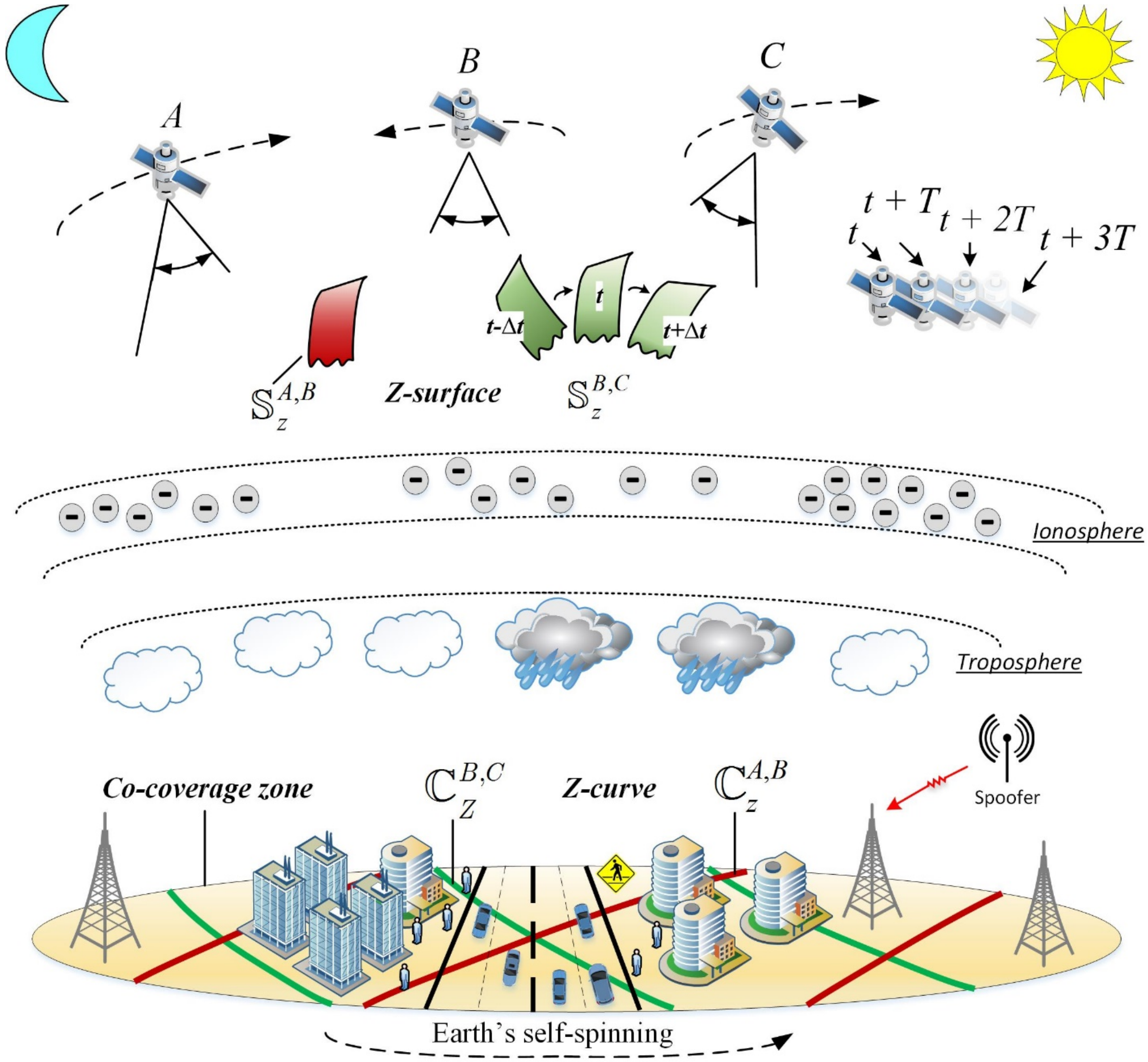


Figure I-4 Illustration of system environment and research tasks

The proposed system of models charactertizes z-curve, z-surface, and the satellite's Terrestrial Service Volume (TSV) at a fixed time, as well as their velocity and morphing patterns over time. A system overview is shown in Figure. For simplicity of technical presentations and when the context is at no risk of losing their generality, I will use parameters of GPS and the ECEF coordinate frame in the following discussions without further elaboration. Specific problems are elaborated as follows:

P1. Formulate the system environment: time, coordinate systems, and the Earth model

Appropriate time and coordinate systems need to be chosen to universally index multiple entities and activities in the subsequent models. The commonly used UTC time (Universal Time, Coordinated) has leap seconds and not continuous. The local clocks on receiver are unstable. Instead, the GPS system time maintained by UTC(USNO) is a continuous time scale synchronized to the UTC and thus will be adopted. Relativistic effects and clock drifts will be compensated accordingly. The Earth-Centered Inertial (ECI) coordinate frame is adequate to describe the motion of satellites, but it is time variant as the Earth spins. In order to describe the co-motion of satellites, the Earth, and receivers, I plan to use the Earth-Centered Earth-Fixed (ECEF) coordinate frame, since most of established models about Earth and ground references are defined in ECEF frame.

P2. Investigate signal processing principle of GNSS receiver, the formations of pseudoranges and z-surface

The z-surface is defined upon pseudorange measurement. The formation of pseudoranges is critical to subsequent modeling process, because signal travels in the speed of light and even a one-microsecond delay would lead to a 300-kilometer error to pseudorange. Receiver's signal processing includes some heavy computation tasks, such as fourier transform and correlation. A non-noticable delay will be introduced in this process, not to mention the delays caused by receiver clock bias, atmospheric delays, and quantization error.

Another issue to be solved in P2 is to select a common timestamp for each z-surface. A z-surface has two pseudoranges and thus four time marks involved. There are 55-65 milliseconds difference between transmited time and received time. Transmitted times from two satellites may also be different. To fix this issue, I assume the pseudorange formation method as the common reception time method [20], such that two pseudoranges will share a same arrival time. This arrival time will be used to index z-surface, and states of satellites will be computed using respective transmitted times after compensation.

P3. Develop a Cartesian geometric model for z-surface at a fixed time

A 3D model for z-surface in the ECEF coordinate frame will be developed. The key challenge is to transform the signal-domain pseudorange relationship into a geometric model. This process needs to be rigorous, to ensure no intermediate step produces extra points that do not satisfy the definition of z-surface. Moreover, some variables may be cancelled out under certain conditions, so I need to carefully make necessary assumptions.

The derived analytic model must be mathematically categorized into some kind of primitive shapes or their combination. Transformation into a canonical form or a proof process may be necessary. With the geometric model, I can then represent the shape using some primitive characteristics, like eccentricity. The expressions of these properties in terms of satellite positions, signal propagation errors, time stamp will be derived. I will use these expressions to profile characteristics of all GPS z-surfaces.

P4. Extend z-surface to k-surface and derive the valid range of k

By selecting different k values, one can have a large number of k-surfaces for his/her system design. I propose to investigate the minimum resolution, the upper and lower bounds for a valid *k* value. The resolution is in proportion to the granularity of pseudorange that a GNSS receiver can identify. For this, I will dig out the answer from Delay Lock Loop (DLL) in a receiver [20-23]. The bounds of *k* are determined by physics of satellite-receiver pair and signal propagations. Too large or too small for *k* values would be physically impossible.

P5. Develop a Cartesian geometric model for TSV at a fixed time

The satellite's TSV are modeled based on a cone, specified by satellite position and radiation beamwidth. Existing TSV models are developed so. However, it is not sufficient to precisely contain all necessary 3D space as the TSV is defined. I propose to add more mathemetical constraints to a basic cone model, for example, excluding the space behind satellites and the dark side of the Earth.

P6. Develop a parameteric model in terms of latitude and longitude for z-curve at a fixed time

The Cartesian model of z-curve has limitations in applications like rendering and discretization. I will propose a parameterization algorithm to trace z-curve using only one parameter, *latitude*. As such, the algorithm can efficiently generate as many points as required by iterating in one-dimension parameter space. The z-curve is essentially a constrained intersection of z-surface (a hyperboloid) and the Earth surface (an ellipsoid), which is categorized as a QSIC or Quadratic Surfaces Intersection Curve. To parameterize it, methods based on numerical approximation are readily available [7][9]. However, an efficient method for *algebraic* solution is rarely studied, but it is highly desirable for precise geodesic and aerospace measurement. I aim to develop an algorithm to compute algebraic coordinates of a z-curve given any latitude of interest.

P7. Develop a parameteric model for satellite footprint at a fixed time

The satellite footprint is also a kind of QSIC, being the intersection of TSV (a cone) and the Earth. Its parameterization algorithm can generate a series of footprint points, which delimits a z-curve. Existing footprint models assume the Earth as a sphere, for simplicity. I propose to directly use the reference ellipsoid for better precision. To deal with the complexity introduced by ellipsoid, I plan to first parameterize the cone into a cluster of rotating lines and then compute the intersection of line versus ellipsoid. The reason is that a cone is a *ruled* surface consisting of infinite many lines, wheras an ellipsoid is not.

P8. Develop migration functions for the TSV and z-surface from time t to t+Δt

With Cartesian models derived by previous tasks, one can give models of TSV and z-surface at time t. This task aims to understand how they are migrating after a time interval Δt. The shapes' dynamics need to be modeled in such a discrete manner, because there are no continuous model that can accurately describe the entire satellite orbit. As illustrated in Figure 4, I will take the updated satellite position at time $t + \Delta t$ as input, and investigate how they will impact the morphing and motion of TSV and z-surface. In the meantime, the signal propagation errors should also be considered. The potential behavior include rotation, translation, scaling, and termination. I am going to apply knowledge of solid geometry to describe these transformations.

P9. Develop a migration function for the z-curve from time t to t+Δt

Z-curve's migration function will help us understand the formation and morphing mechanism of ground areas that are segmented by the combination of k-curves. It may also reduce the computations of regenerating a new z-curve at time *t*+Δ*t*, which is a costly computational task to calculate the QSIC. Z-curve migrates along with the z-surface it belongs to. However, the migration function of z-surface cannot be applied directly to z-curve, because the z-curve may become off the Earth surface after transformation at time *t*+Δ*t*. Since it is relatively hard to obtain an algebraic solution, I propose a migration function based on numerical calculations.

I.4 Organizations of This Dissertation

The remainder of this dissertation is organized as follows. In Chapter II, I will discuss the starter of this research – recognizing the z-surface from the perspective of an individual receiver. In Chapter III, I begin to formalize the models of z-surface at a fixed time, and to develop the parameterization algorithm for the static z-curve. In Chapter IV, I discuss the geometric model for a satellite's TSV, and the parameterization of its coverage footprint. In Chapter V, I analyze the motion behavior of z-surface, TSV, and z-curve.

CHAPTER II

GNSS DISCRETE EVENT – POINT BASED ORDER OF PSEUDORNAGES[1]

II.1 GNSS Discrete Events

Correct ordering of concurrent system events, e.g., "sensor reading A happened *before* actuator event B", or vice versa, is essential to guarantee deterministic outcomes of distributed computing. This requirement affects essentially all layers of system abstractions, from the basic task of interpretation of system states, to high level semantics of system behaviors. In his seminal work entitled "Time, Clocks, and the Ordering of Events in a Distributed System" [1], Lamport formalized the relationship between time, clocks and ordering of concurrent events among networking nodes which run on their own high speed clocks. To resolve the timing ambiguity of concurrent messages due to network delays, Lamport proposed the notion of message based *virtual clocks* for interacting nodes to reason about orders of concurrent events. As a result, the rate of the virtual clock, not that of local clocks, defines the highest rate of distribution interactions with guaranteed outcomes. His work is credited for laying a key theoretical foundation to prove correctness of networking protocols and distributed algorithms, e.g., Internet protocols, Domain Name System (DNS), distributed databases, etc.

As the GNSS technology rapidly expands from the standalone receiver architecture to networked environments, how to guarantee correct use of the time and position information for broad applications is an emerging issue. Different from the assumption of independent clocks in

[11] Reprinted from "Order of Pseudoranges (OOP) of GNSS: Spatial modeling and analysis" by Guoyu Fu and Jyh-Charn Liu in *Proceedings of the 28th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, pp. 3370-3382. 2015. Copyright 2015 by Guoyu Fu.

system nodes [1], GNSS systems are based on a master clock network (ground control and space segments) to generate the highly accurate timing signals for space based broadcast. For instance, the GPS system time is an extremely precise global time reference aligned with the *UTC(USNO)* or *Coordinated Universal Time (U. S. Naval Observatory)*, which is generated and maintained from a network of atomic clocks. How to harmonize the GNSS clocking-timing model with the classical distributed system architectures is crucial to the development of the future generation of GNSS based systems. Correct ordering of GNSS signals based events is essential to guarantee deterministic outcomes when GNSS signals are an integrated part of distributed computing systems.

The clocking sources, transmission delays of the atmosphere, astrophysical effects on the clocking subsystems, as well as algorithms for receivers to recover the GNSS time have been thoroughly studied for decades. Yet, how to leverage on these high quality signals to create event ordering abstractions in the temporal and spatial domains is still a largely untouched issue. Knowing that pseudorange is the lowest level of computational abstraction for a receiver to acquire its local time and position, I propose the notion of *order of pseudoranges* (OOP) as a type of GNSS system states to create GNSS system ordering events. OOP refers to a sorted order of pseudoranges, which represent their measured distances to satellites. Receivers within proximity of each other should have the same OOP, but key questions are when and where the OOP changes. The OOP changes from one state to another when the order of two or more pseudoranges needs to be shuffled. I call such a transition a *zero-crossing event* because the ordering relationship happens in the sequence of $\rho_A - \rho_B > 0$: $\rho_A - \rho_B = 0$: $\rho_A - \rho_B < 0$, where $\rho_A(\rho_B)$ denotes the pseudorange for satellite A(B).

Due to its complexity, this chapter only considers the spatial properties of OOP at a snapshot of time. I formalize and derive key properties of OOP. Such properties can be used to create a wide area reference grid for event ordering. The discussion starts with a simple model of RF signal propagations from satellites to their covered areas. The intersection of the covered areas for a set of satellites forms their co-coverage area. Positions that simultaneously observe 2 or more identical pseudoranges form the natural boundary surface (called *z-surface*) between geospatial areas defined by orders of pseudoranges. The intersecting curve of a z-surface and the Earth surface is called a *z-curve*. Defined by a set of intersecting z-curves, an *O-zone* is a ground region where all receivers have the same OOP state. The high speed flights of satellites (~4000 meters/second) lead to fast sweeping of z-curves on the ground at rate of hundreds of meters per second. As a result, the shape and size of O-zones also morph with time.

Using the log data of the CORS (Continuously Operating Reference Stations) network, I test the OOP model to show its accuracy within the resolution of the CORS receivers. A direct application of OOP is design of wide-area architecture for timing integrity check of GNSS signals. Novel applications and computing abstractions can be developed on the basis of OOP, e.g., location based routing, wide area check point for GNSS time integrity check, wide area sensing and measurement control, or GNSS based coordination and time stamping in wide area distributed computing systems, etc.

The remainder of this chapter is organized as follows. Section II.2 highlights the system environment. Sections II.3, II.4, and II.5 describe the pseudorange model, OOP model, and the relationship of a single receiver to z-surface, respectively. Section II.6 illustrates experimental results with measurement data from the CORS receivers.

II.2 System Overview

A GPS satellite has a radiation beam with width slightly larger than 20º [26]. Referring to Fig. 1, circle-like area on the Earth surface represents the coverage zone of a satellite s, denoted as R_s^C. Following the typical model on GPS signal transmission, I assume that a receiver u can receive the signals from a satellite s when the elevation angle of s to u is greater than 5º [27], e.g., the receiver marked by a green-dot in Figure II-1. A co-coverage zone R_S^C of a set of satellites S is defined if all points on this surficial area are visible to all satellites in S. For instance, in Figure II-1, the co-coverage zone for the two satellites is the area with purple boundary line. Through a simple deductive process, it is straightforward to prove that the co-coverage zone is convex, and not fragmented.

$$R_S^C = \bigcap_{\forall s_i \in S} R_{s_i}^C \tag{II.1}$$

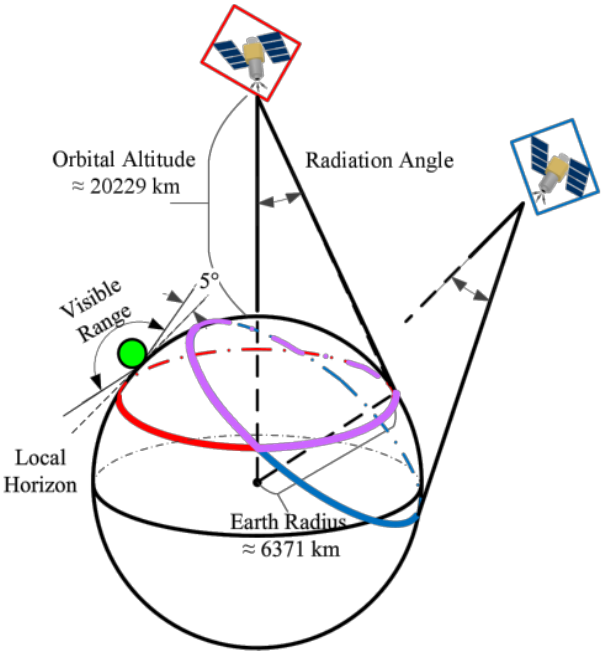


Figure II-1 Co-coverage of two satellites

The average orbital altitude of GPS satellites (20229 kilometers) is about three times of the Earth radius (6371 kilometers) [26]. As a result, the z-curve between two satellites is not necessarily on the Earth surface. When they do, their z-curves may or may not be located within the co-coverage zone for a particular set of satellites of interest. That being said, overall a large number of z-surfaces are available for the GPS system. Table II-1 illustrates the number of z-surfaces formed from a 24-satellite GPS constellation [26]. For more than 30% of time over the globe, there will be up to 28 z-surfaces that can be seen in a co-coverage zone of 8 satellites. This number will increase significantly, as more satellites are being planned for launching.

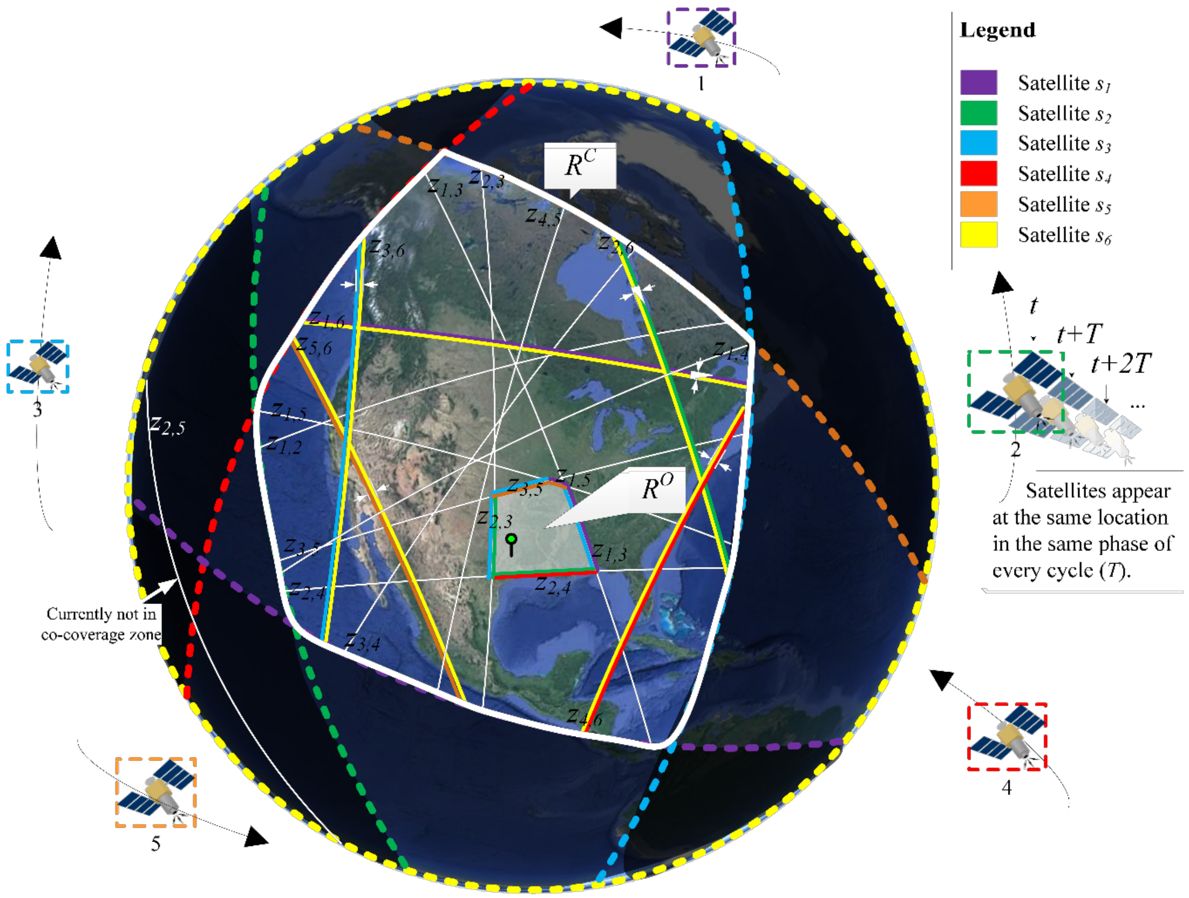


Figure II-2 Overview of North America (viewed from the sixth satellite)

Next, I use the co-coverage area of 6 randomly positioned satellites s_{1-6} to discuss their OOP on the earth surface. The artwork shown in Figure II-2 is used to facilitate the discussion, where s_{1-5} are drawn at the outskirt of Figure II-2, and s_6 is placed at the reader's eye position (and thus not explicitly drawn). This way, the figure and the reader form a 2½-dimensional perceptional space for discussion of RF signal transmissions to their co-coverage zone. The boundary of the coverage zone is plotted as color-coded dash lines, where the satellite color codes are listed in the legend. The coverage zone $R_{s_6}^{C}$ is the black circle area with the yellow dash line boundary. The co-coverage zone of the 6 satellites $R_S^C = \bigcap_{i=1,2,\ldots,6} R_{s_i}^{C}$ is illustrated as the area defined by the white thick surface polygon at the center. Within R_S^C, a receiver can measure its pseudoranges from the 6 satellites. A few (not to scale) z-curves, which are marked in two colors representing involved satellites, are highlighted to represent the equal-pseudorange curve of two satellites. For instance, the green-red line segment with the label of $z_{2,4}$ represents the z-curve between s_2 and s_4. Receivers located at the green (red) side of the line have shorter pseudoranges to s_2 (s_4). The remaining z-curves are represented by white lines.

Table II-1 The estimated probability of the number of satellites in view and their number of z-curves [26].

# satellites in view	5	6	7	8	9	10	11	12
Probability (%)	0.01-0.1	1-10	10-30	>30	10-30	1-10	0.1-1	0.001-0.1
# *z*-surfaces	10	15	21	28	36	45	55	66
# *O*-zones	120	720	5040	40320	362880	3628800	39916800	479001600

The OOP_u is simply the (ascending) order of pseudoranges of a receiver at a position u to a (sub)set of satellites in its view. For instance, when the OOP_u is defined for two satellites s_i and s_j(i and j are satellite numbers), $OOP_x = [i, j]$ when $\rho_i < \rho_j$, where ρ_i denotes the pseudorange to s_i. The receiver observes crossing of a $z_{i,j}$-surface (or $z_{i,j}$-curve) of s_i and s_j when $\rho_i = \rho_j$. The solution of the pseudoranges of equal value may be subject to errors in ephemeris data or other nominal noise. The uncertainty in estimation of these nominal errors is translated to the width of a z-curve. In order to have a focused discussion, we assume that all these nominal errors to be known during modeling process, and the width of z-curve is assumed to be zero in the geometric modeling process of OOP.

Each OOP configuration is associated with a three-dimensional (3D) space bounded by z-surfaces, where all points (receivers) inside a zone should have the same order of pseudoranges. Some of them intersect with the Earth surface, forming O-zones bounded by z-curves. As an example illustrated in Figure II-2, the receiver (green dot) is surrounded by several z-curves nearby, i.e. $z_{1,5}$, $z_{1,3}$, $z_{2,3}$, $z_{2,4}$ and $z_{3,5}$, and by some z-curves in farther distance, i.e. $z_{1,2}$, $z_{1,6}$, $z_{3,4}$, $z_{4,5}$, etc. Based on the definition of z-curves, the receiver knows that its order of pseudoranges should be $OOP_u = [6, 2, 4, 5, 3,1]$, which is the signature of the white shaded zone (R^O).

II.3 Pseudorange Model

The subsequent discussions fix the time advancement and assume that pseudoranges are measured during the same short period of time ($\leq 2\epsilon$) as defined in (II.2). The pseudorange $\rho_s(\overline{r^u})$to satellite s is a computed approximation of their distances $\|\overline{r^s} - \overline{r^u}\|$. The source of error in ρ can be categorized into *nominal errors* $E1$ and *unexpected error(s)* $E2$. $E1$ is the algebraic sum of ionospheric delay δ^{iono}, tropospheric delay δ^{trop}, satellite clock bias δ^{sc} and user clock bias δ^{uc} [25][26][28]. Other small errors with 1-meter or below root-mean-square (rms) magnitudes, e.g., receiver noise, are ignored [26]. A formal description of their relationship is given as follows:

$$\rho = \|\overline{r^s} - \overline{r^u}\| + cE1 + cE2, \tag{II.2}$$

$$E1 = \delta^{iono} + \delta^{trop} - \delta^{sc} + \delta^{uc}, \tag{II.3}$$

where c represents the speed of light in vacuum. When $E2$ exceeds certain range, e.g., when caused by malfunction of the receiver, spoofing [40][41][125], etc., the measured pseudorange will exceed a threshold value that is considered to be reasonable. That is to say, an affected pseudorange will lead the receiver to an incorrect timing of zero-crossing event. As it will be clear later, the range of reasonable errors can be translated as the duration of a zero-crossing event. To have a more focused discussion, I assume $E2 = 0$ here.

The magnitudes of δ's in $E1$ are known to be within certain bounds. Atmospheric delays, i.e. δ^{iono} and δ^{trop}, are caused by interference of the widely homogeneous ionosphere and troposphere above a broad area of receivers [28]. Satellite clock bias δ^{sc} is inherent in broadcast signals and affects all receivers to the same amount in coverage zone. The OOP model benefits from the spatial locality (homogeneity) of atmospheric delays and satellite clock drifts, in a way similar to that of Differential GPS (D-GPS) [26]. The receiver clock bias δ^{uc} has the largest magnitude among different $E1$ error terms [28]. It refers to the difference between the *GPS system time* and the reading of receiver clock. As it will be clear later that in the pairwise OOP model, receiver clock bias is cancelled out when one takes the difference between two pseudoranges for comparison. Being able to eliminate the largest error term greatly reduces the range of uncertainty in OOP modeling and analysis. For the remaining $E1$ errors, $c\delta^{iono}, c\delta^{trop}$ and $c\delta^{sc}$ has non-negligible error size. For example, the satellite clock bias can reach the range of 1 millisecond, which will translate to 300-kilometer pseudorange error [27]. Conventional models or DGPS techniques could help reduce the error size down to 2-30 meters, 2-25 meters and 2 meters in rms, with respect to $c\delta^{iono}, c\delta^{trop}$ and $c\delta^{sc}$, respectively [26]. Statistical distributions of $E1$ errors will help determine the reasonableness of measurement errors.

II.4 Pairwise OOP Model

The OOP model can be defined for an arbitrary number of satellites. In this subsection, we discuss the OOP model for a pair of satellites s_i and s_j (let $i < j$ without loss of generality). Let a Z function, $z_{i,j}(\overline{r^u}) = \rho_j(\overline{r^u}) - \rho_i(\overline{r^u})$, denotes the difference of pseudoranges ρ_i and ρ_j (measured in the same 2ϵ period). As it will be clear later, when Z is equal to zero, the 3D solution surface in terms of receiver positions is the *hyperboloid* zero-crossing surface (called z-surface in short) as shown in Figure II-3, which is denoted as $z_{i,j} = \{\overline{r^{u*}} \mid z_{i,j}(\overline{r^{u*}}) = 0\}$, where $\overline{r^{u*}}$ refers to positions satisfying the equation. I claim that a receiver u experiences a zero-crossing event when u is situated on the z-surface. u should measure change of the order for ρ_i and ρ_j after the zero-crossing event of $z_{i,j}$-surface. The distance of u to the z-surface (denoted as d^z) can be used to determine its relative location to the z-surface. The relationship between z-surface to receiver u, such as d^z, is analyzed with the aid of the 3D symmetric plane (called h-plane) between two satellites, as illustrated in the Figure II-3.

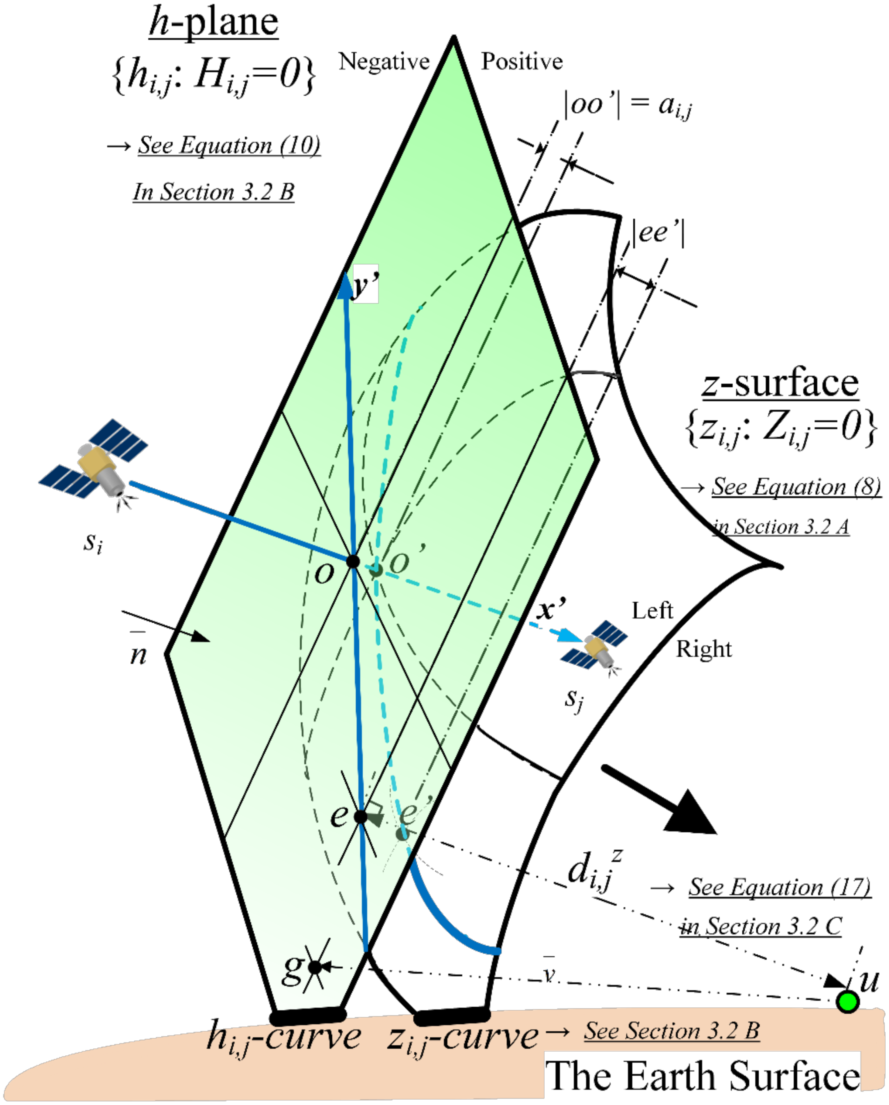


Figure II-3 Pairwise OOP model

The h-plane is helpful to reduce the complexity of mathematical expression and to visualize the geometry and motion dynamics. I define the H function as the difference of distances from receiver to s_i and s_j, or $H_{i,j}(\overline{r^u}) = \|\overline{r_j^s} - \overline{r^u}\| - \|\overline{r_i^s} - \overline{r^u}\|$. The h-plane is then the solution plane in terms of receiver positions, denoted as $h_{i,j} = \{\overline{r^{u*}} \mid H_{i,j}(\overline{r^{u*}}) = 0\}$. Receivers on the h-plane have the same distance to the two satellites. The h-plane and z-surface intersect with the Earth ground respectively to form their h-curve and z-curve, as illustrated in Figure II-3.

The Z Function and Z-Surface

Ordering of ρ_i and ρ_j can be determined by the $Z_{i,j}$ function:

$$\begin{cases} \rho_i < \rho_j, & if\ Z_{i,j} > 0, \\ \rho_i = \rho_j, & if\ Z_{i,j} = 0, \\ \rho_i > \rho_j, & if\ Z_{i,j} < 0. \end{cases} \tag{II.4}$$

To model the relationship in terms of receiver's position $\overline{r^u} = (x, y, z)^T$, I expand function $Z_{i,j}(\overline{r^u})$ according to (I.3) and (I.4):

$$Z_{i,j}(\overline{r^u}) = \left\|\overline{r_j^s} - \overline{r^u}\right\| - \left\|\overline{r_i^s} - \overline{r^u}\right\| + 2a_{i,j}, \tag{II.5}$$

where $\overline{r_i^s} = (x_i, y_i, z_i)^T$ and $\overline{r_j^s} = (x_j, y_j, z_j)^T$. The expression $a_{i,j} = \frac{cE1_j - cE1_i}{2}$ is called the *semi-major axis*, and

$$a_{i,j} = \frac{c(\delta_j^{iono} + \delta_j^{trop} - \delta_j^{sc}) - c(\delta_i^{iono} + \delta_i^{trop} - \delta_i^{sc})}{2}. \tag{II.6}$$

Note that $\delta_i^{uc} = \delta_j^{uc}$ are in the same 2ϵ period and they are cancelled out. $a_{i,j}$ is half of difference between nominal errors from two channels s_i and s_j. Because remaining nominal errors (δ^{iono}, δ^{trop} and δ^{sc}) have a strong spatial homogeneity, I assume that a is a function of time alone, and is invariant to the receiver positions.

The zero-crossing surface $z_{i,j}$ of s_i and s_j is the solution space of the equation $\{\overline{r^{u*}} \mid Z_{i,j}(\overline{r^{u*}}) = 0\}$. That is, every receiver at $\overline{r^{u*}}$ on the $z_{i,j}$-surface has pseudoranges $\rho_i(\overline{r^{u*}}) = \rho_j(\overline{r^{u*}})$, according to (II.5). The satellites s_i and s_j are two foci of surface $z_{i,j}$. The surface separates the 3D space into two spaces, called *left zone* and *right zone*. A receiver in left zone, in which $Z_{i,j} > 0$, measures that $\rho_j > \rho_i$. Otherwise, in right zone, it measures $\rho_i > \rho_j$.

When a receiver passed across $z_{i,j}$-surface, it should detect a change of the order between ρ_i and ρ_j. It will become clear shortly that one of the left/right zones is convex while the other concave.

The $\{z_{i,j}: Z_{i,j}(\overline{r^{u*}}) = 0\}$ equation can be further expanded as:

$$z_{i,j}: \sqrt{\left(x_j - x\right)^2 + \left(y_j - y\right)^2 + \left(z_j - z\right)^2} - \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = 2a_{i,j}, \tag{II.7}$$

where $a_{i,j}$ is an arbitrary constant value. From (II.7), we observe that the distance from a point to s_j minus that to s_i remains unchanged the position of the point. Recall that a 2D hyperbola is a set of points where the absolute value of the difference of the distances to the foci is a constant [126]. For $z_{i,j}$, traces of moving points under the hyperbola constraint by (II.7) on any fixed 2D plane that joins two satellites (for example, the 2D coordinate frame $ox'y'$ in Figure II-3, as will be defined later), form one branch of the hyperbola. By rotating the 2D plane 180º around the major axis connecting two foci, one can observe one sheet of a hyperboloid in the 3D space, which is exactly the $z_{i,j}$-surface.

In the hyperbola definition, the constant difference of two distances is twice the length from the center to the vertex of each branch. Accordingly, we model the distance from the center (point o) of two foci to the vertex (point o') of a sheet of hyperboloid as the semi-major axis $a_{i,j}$, as illustrated in Figure II-3. Different from general hyperboloid, equation (II.7) does not describe the absolute difference of two distances so it represents only one sheet of hyperboloid. In Figure II-3, I set $a_{i,j}$ to be less than zero and thus the right sheet near s_j is drawn. Points in the left zone (concave) are slightly more than those in the right zone (convex). If $a_{i,j} > 0$, the $z_{i,j}$-surface will be the left sheet of hyperboloid that is near s_i. The curvature of the $z_{i,j}$-surface as well as the offset to $h_{i,j}$ increases with the value of $|a_{i,j}|$. In practice, $|a_{i,j}|$ is relatively small compared to the distance of two satellites. As such, for z-surface analysis, it is helpful to start with the extreme case that when $a_{i,j}$ is equal to zero, the $z_{i,j}$-surface collapses down as the $h_{i,j}$-plane. As a preamble to the analysis of the relationship between z-surface and receiver, next I discuss modeling of the h-plane.

The h-Plane

The equal-distance plane $h_{i,j}$ of s_i and s_j is the solution space of $\{\overline{r^{u*}} \mid H_{i,j}(\overline{r^{u*}}) = 0\}$, or formally:

$$h_{i,j}: \sqrt{\left(x_j - x\right)^2 + \left(y_j - y\right)^2 + \left(z_j - z\right)^2} = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2}. \quad \text{(II.8)}$$

All points on $h_{i,j}$ have equal distances to two foci s_i and s_j. The center point is $\bar{o} = (\frac{x_i+x_j}{2}, \frac{y_i+y_j}{2}, \frac{z_i+z_j}{2})^T$. The space of points which are near s_j have is called the *positive space* (because $\rho_j > \rho_i$), while the other space is the *negative space*. Now that $h_{i,j}$ is a plane in 3D space, it has the form:

$$h_{i,j}: A_{i,j}x + B_{i,j}y + C_{i,j}z + D_{i,j} = 0, \quad \text{(II.9)}$$

where A, B, C and D are time-dependent variables but position-invariant constants. Based on the basic geometry of the symmetry plane between two points [127], one can directly obtain expressions of the four parameters as follows:

$$A_{i,j} = x_j - x_i, \quad \text{(II.101)}$$

$$B_{i,j} = y_j - y_i, \quad \text{(II.11)}$$

$$C_{i,j} = z_j - z_i, \quad \text{(II.12)}$$

$$D_{i,j} = -\overline{n}^T\overline{O}, \quad \text{(II.13)}$$

where $\overline{n}$ is the normal vector of the plane $h_{i,j}$, indicating its direction from s_i to s_j, or formally:

$$\overline{n_{i,j}} = \left(x_j - x_i, y_j - y_i, z_j - z_i\right)^T. \qquad \text{(II.14)}$$

Therefore, one can construct $h_{i,j}$-plane of any two satellites using only their positions $\overline{r_i^s}$ and $\overline{r_j^s}$, which can be obtained through many precise resources, such as the public ephemeris model [27].

I make z-curves (as drawn in Figure II-2) shape like ellipse segments, as it is near the h-curve. It is well known that the intersection of a 3D plane and an ellipsoid (the Earth surface) is an ellipse [128]. Given the formula of the intersecting ellipse (as the h-curve illustrated in Fig. 3), it is easier to derive the closed-form equation of z-curve. Theoretically, the closed-form equation for h-curve could be obtained by solving the system of equation (10) and the following Earth reference ellipsoid equation [18]:

$$\frac{x^2}{a_w^2} + \frac{y^2}{a_w^2} + \frac{z^2}{b_w^2} = 1, \qquad \text{(II.15)}$$

where a_w and b_w are known constants defined by the World Geodetic System 84 (WGS84) [18]. Unfortunately, there is not an explicit and simple solution for the system of equations (I.10) and (I.16) [128][129]. It is even more difficult to derive a straightforward formula for the z-curve, not to mention its relation with the co-coverage zone boundary. As an alternative, numerical methods could be applied to simulated or measured OOP data, e.g., data from the CORS network, to derive the z-curve. The numerical method has an advantage over the analytic method, because the latter only takes the approximate Earth surface into account.

Provided with the mathematic equation of z-curve, its relations to ground receivers and O-zone divisions can be computed in an explicit way. Since it is not available, I put our emphasis on the 3D z-surface to model its characteristics, from the receiver u's perspective.

II.5 The Z-Surface's Relations to Receiver

For a receiver u at a known position $\overline{r^u}$, questions of interest about the every z-surface include but are not limited to:

- Is u locating at left zone or right zone of an z-surface?
- What is the distance from u to the z-surface?
- Is the surface moving away or towards u?
- If it is getting closer, when will u experience next zero-crossing event?

To answer these questions, the distance from u to the z-surface, denoted as $d^z\ (\overline{r^u})$, is the key. I define this distance as illustrated in Figure II-3: the length of the line segment ue', where e' is a point on z-surface intersecting with the perpendicular line ue from point u to the point e on h-plane. Let $d^z_{i,j}(\overline{r^u})$, $d^h_{i,j}(\overline{r^u})$ denote the length of $|ue'|$ and $|ue|$, respectively. As it will be clear later, when $d^z_{i,j}(\overline{r^u}) > 0$, the receiver is located at the right zone near s_j, otherwise, at the left near s_i. When $d^h_{i,j}(\overline{r^u}) > 0$, it is located at the positive side of h-plane, otherwise, at the negative side, as shown in Figure II-3.

To analyze $d^z_{i,j}(\overline{r^u})$, I construct a 2D Cartesian coordinate frame $ox'y'$ with the origin at point o, the x'-axis from s_i to s_j and the y'-axis from point e to o, as shown in Figure II-3. By definition, points s_i, s_j, o and o' are on the new coordinate plane $ox'y'$. I claim that point e' is also on the coordinate plane $ox'y'$, which is proved through this process ("$\Longrightarrow$" means "lead to"): segment ue and the x'-axis are both perpendicular to the plane $\Longrightarrow$ they are in parallel, which forms a plane containing s_i, s_j, o and o' $\Longrightarrow$ all points on ue are also on the plane $\Longrightarrow$ point e' is on the plane.

As plane $ox'y'$ being the cross profile, the h-plane and z-surface become a line and a hyperbola, respectively. Depending on the position of u, the sign of a and the length of $|ee'|$, eight different relations among $d^z_{i,j}(\overline{r^u})$ components are shown in Table II-2. The bottom four cases illustrate when point u is within the furrow between the line and the hyperbola, i.e. $|d^h| < |ee'|$. Let $sign()$ denote the sign function which returns 1, -1 or 0 when the input is positive, negative or zero, respectively. The relationship between d^h, d^z, a and $|ee'|$ can be summarized from Table II-2 as the following equation:

$$d^z_{i,j}(\overline{r^u}) = d^h_{i,j}(\overline{r^u}) + sign(a_{i,j}) \cdot |ee'|. \tag{II.16}$$

It is important to note that $d^z_{i,j}(\overline{r^u}) = 0$ is a necessary, but not sufficient condition for zero-crossing events, because it only indicates the $z_{i,j}$-surface is flying across u. It doesn't imply that both s_i and s_j are visible to u. Therefore, another necessary condition for zero-crossing event is that u is located within the co-coverage zone $R^C_{s_i,s_j}$.

In (II.17), $d^h_{i,j}$ and $|ee'|$ can be computed with satellite positions and receiver position. $d^h_{i,j}(\overline{r^u})$ is the distance of a point to a 3D plane. The conventional equation [127] is revised to show the *sign of distance*, because it indicates which side (positive or negative) the receiver u is in. $d^h_{i,j}(\overline{r^u})$ is given as follows:

$$d^h_{i,j}(\overline{r^u}) = \frac{A_{i,j}x + B_{i,j}y + C_{i,j}z + D_{i,j}}{\left|\overline{n_{i,j}}\right|}, \tag{II.17}$$

where $\overline{n_{i,j}} = (A_{i,j}, B_{i,j}, C_{i,j})^T$ is the normal vector of plane $h_{i,j}$, pointing from s_i to s_j as defined in (15). $d^h_{i,j}(\overline{r^u}) > 0$ (or <) is equivalent to that the point u is on the positive (or negative) side of the plane $h_{i,j}$. One can reason about this claim as follows: let $\bar{v}$ denote a vector from any point g

on the plane to u (as shown in Figure II-3), (II.18) is derived from the length of projected component of $\bar{v}$ onto the direction of $\overline{n_{i,j}}$, or $d_{i,j}^{h}(\overline{r^{u}}) = \frac{\bar{v}^{T}\overline{n_{i,j}}}{\left|\overline{n_{i,j}}\right|}$.

To analyze $|ee'|$, we focus on the plane $ox'y'$. Because ee' is perpendicular to the y'-axis, $|ee'|$ is the x' component of point e' coordinate. As such, the 2D equation for the hyperbola $o'e'$ is needed. The general form for any hyperbola whose transverse axis is aligned with the x'-axis and center is on the origin o, is expressed as [126]:

$$\frac{x'^2}{a'^2} - \frac{y'^2}{b'^2} = 1, \tag{II.18}$$

where ${a'}^2 = a_{i,j}^2$ obviously, b' is the length of the conjugate axis. According to basic hyperbola knowledge, we can derive the equation of b'^2 [126]:

$$b'^2 = \frac{|s_i s_j|^2}{4} - a_{i,j}^2, \tag{II.19}$$

where $|s_i s_j|$ is the distance between two focal satellites. Then the coordinate of point e' in the $ox'y'$ frame is:

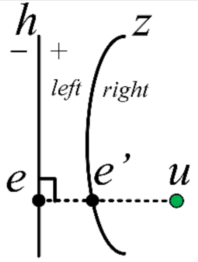
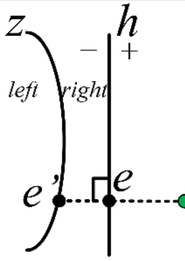
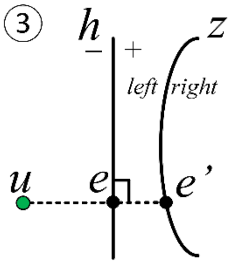
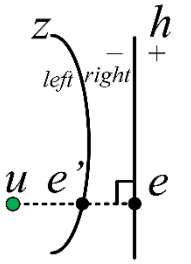
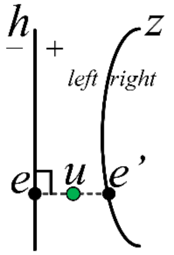
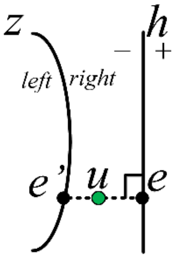
$$\bar{e'} = \left[-sign(a_{i,j}) \cdot \sqrt{a_{i,j}^2\left(1 + \frac{4 \cdot |oe|^2}{\left|s_i s_j\right|^2 - 4a_{i,j}^2}\right)}, -|oe|\right], \tag{II.20}$$

where $|oe|$ can be obtained through the right triangle Δoeu, i.e. $|oe| = \sqrt{|ou|^2 - {d_{i,j}^{h}}^2}$.

Therefore, the length of segment ee' is:

$$|ee'| = \sqrt{a_{i,j}^2\left(1 + \frac{4 \cdot \left(|ou|^2 - {d_{i,j}^{h}}^2\right)}{\left|s_i s_j\right|^2 - 4a_{i,j}^2}\right)}. \tag{II.21}$$

Table II-2 Relationship of d^z, d^h and $|ee'|$ in cases with different relative positions of u, h-plane and z-surface.

$\lvert d^h\rvert > \lvert ee'\rvert$	$a_{i,j} < 0$	$a_{i,j} > 0$
$d^h > 0$	①  $d^z=d^h-\lvert ee'\rvert$	②  $d^z=d^h+\lvert ee'\rvert$
$d^h < 0$	③  $d^z=d^h-\lvert ee'\rvert$	④  $d^z=d^h+\lvert ee'\rvert$
$\lvert d^h\rvert < \lvert ee'\rvert$	$a_{i,j} < 0$	$a_{i,j} > 0$
$d^h > 0$	⑤  $d^z=d^h-\lvert ee'\rvert$	⑥ same with case 2
$d^h < 0$	⑦ same with case 3	⑧  $d^z=d^h+\lvert ee'\rvert$

To get a sense of the magnitude of $|ee'|$, we list some typical values in the following table as the $|a_{i,j}|$ and $|s_i s_j|$ change. $|oe|$ is fixed to be 20229 kilometers, which is the average orbital altitude 20229 km [26].

Table II-3 Typical values of |ee'| (kilometers)

$\boldsymbol{\lvert ee' \rvert}$		$\boldsymbol{\lvert s_i s_j \rvert}$			
		1,000	**5,000**	**10,000**	**30,000**
$\boldsymbol{\lvert a_{i,j} \rvert}$	**5**	202	41	21	8
	20	810	163	83	34
	50	2033	408	208	84
	100	4130	816	417	168

From Table II-3, the value of $|ee'|$ will be amplified as two satellites fly closer to each other and as the value of $|a|$ gets larger. The curvature of z-surface is significant since the hyperboloid sheet expands significantly as it makes contact with the ground. Fortunately, $a_{i,j}$ can be effectively estimated based on many well established error correction models [27][37][38][39] and techniques (DGPS) [26][28]. The estimation uncertainty remains to be the width of the z-surface where the actual hyperboloid could be any sheet in it.

To answer Q1 and Q2, a receiver first estimates $a_{i,j}$ using correction models or DGPS corrections, and then compute $d^z\ (\overline{r^u})$ using (II.17), (II.18) and (II.22). $d^z\ (\overline{r^u})$ is more explicit and simpler to compute than using the shortest distance from u to the z-surface. For $d^z\ (\overline{r^u})$, the point on surface e' is coupled with the h-plane and is easy to be tracked as the h-plane is dragged by the satellites, and it does not cause loss of precision. For the shortest distance, one can compute it using methods like Lagrange multipliers [126], but it would require complex calculation process and high computing cost for iterative solution of the system of equations. The two approaches eventually converge to each other, because a zero-crossing event happens if and only if both $d^z\ (\overline{r^u})$ and the shortest distance are equal to zero.

Q3 and Q4 are related to analysis of point e at change of $d^z_{i,j}$ over time. Dynamics of point e on the h-surface is complex because it is affected by the motion pattern of satellites. The h-plane on the z-surface is "dragged" by two orbiting satellites on foci at their respective velocities. The movement of the center point o is the averaged motion of two satellites. However, other points on the h-plane have different angular and linear speeds, since the plane is not spinning around some center. Notice that the receiver positioning algorithm regularly provides computation results of satellite positions using current ephemeris. Therefore, I can analyze the dynamics of $d^z_{i,j}$ discretely by considering consecutive two sample times t_{k-1} and t_k, where k is a positive integer.

Let $v^z_{i,j}$ denote the rate of $d^z_{i,j}$ over time, which can be approximated through:

$$v^z_{i,j}(\overline{r^u}, t_k) = \frac{d^z_{i,j}(\overline{r^u}(t_k), t_k) - d^z_{i,j}(\overline{r^u}(t_{k-1}), t_{k-1})}{t_k - t_{k-1}}. \qquad \text{(II.22)}$$

From (II.17) and (II.23), one can know that the z-surface is fading away from the z-surface when $sign(d^z_{i,j})sign(v^z_{i,j}) > 0$ and thus it is less urgent to keep track of this pair of satellites. The receiver is approaching the z-surface if $sign(d^z_{i,j})sign(v^z_{i,j}) < 0$ because in such scenario $d^z_{i,j}$ could decrease until $d^z_{i,j} = 0$, implying an imminent zero-crossing event. The time left to cross the $z_{i,j}$-surface, denoted as $T^z_{i,j}(t_k)$, is defined as the difference of current time t_k to the zero-crossing time $t^z_{i,j}$, or $T^z_{i,j}(t_k) = t^z_{i,j} - t_k$. Formally, it can be approximated by:

$$T_{i,j}^{z}(\overline{r^{u}}, t_k) = \frac{\left|d_{i,j}^{z}(\overline{r^{u}}, t_k)\right|}{\left|v_{i,j}^{z}(\overline{r^{u}}, t_k)\right|}, if \;\; sign\left(d_{i,j}^{z}\right)sign\left(v_{i,j}^{z}\right) < 0 \,. \tag{II.23}$$

To answer Q3 and Q4, a receiver needs to do the following steps:

1. Compute and store current and previous samples using (II.17), (II.18) and (II.22), i.e. $d_{i,j}^{z}(\overline{r^{u}}(t_k), t_k)$ and $d_{i,j}^{z}(\overline{r^{u}}(t_{k-1}), t_{k-1})$;
2. Calculate the velocity $v_{i,j}^{z}(\overline{r^{u}}, t_k)$ using (II.22) to determine the relative motion of z-surface;
3. If the surface is approach, one can predict $T_{i,j}^{z}(t_k)$ using (II.24).

II.6 Experimental Validation and Analysis

The performance of the OOP model is evaluated using the publically available CORS measurement data [23]. The 1-second RINEX (Receiver Independent Exchange Format) [130] files of GPS data from CORS locations marked on Figure II-4, JPLM, ICT1, ICT2, ICT4 and ICT5 were used for evaluation of the OOP model. Raw L1 code pseudoranges are recorded in the RINEX files. For indexing of satellite pairs for the 32 GPS satellites, we design a simple formula for a total of 496 labels (1, 2), (1, 3), …, (1, 32), (2, 3), …, (31, 32) as follows:

$$Index(i,j) = \frac{(64-i)(i-1)}{2} + j - i \tag{II.24}$$

A zero-crossing event occurred when the order of two pseudoranges swapped. Figure II-5 plots pseudoranges measured by JPLM to all GPS satellites over 86400 seconds in the 145th day of 2015. Pseudorange from the same satellites is marked in the same color. The intersection of any two curves of different colors is a zero-crossing point. In the histogram format, Figure II-6 shows the number of zero-crossing events observed at JPLM. The frequency and distribution of these zero-crossing points are determined by the position of JPLM with respect to the GPS constellation. A potential application of OOP is to use these zero-crossing points as timing integrity check points based on matching satellite pairs' behaviors. When JPLM is used alone, the checking frequency is in the interval of 15-30 minutes. This checking frequency can be drastically increased with multiple stations are networked to serve this purpose.

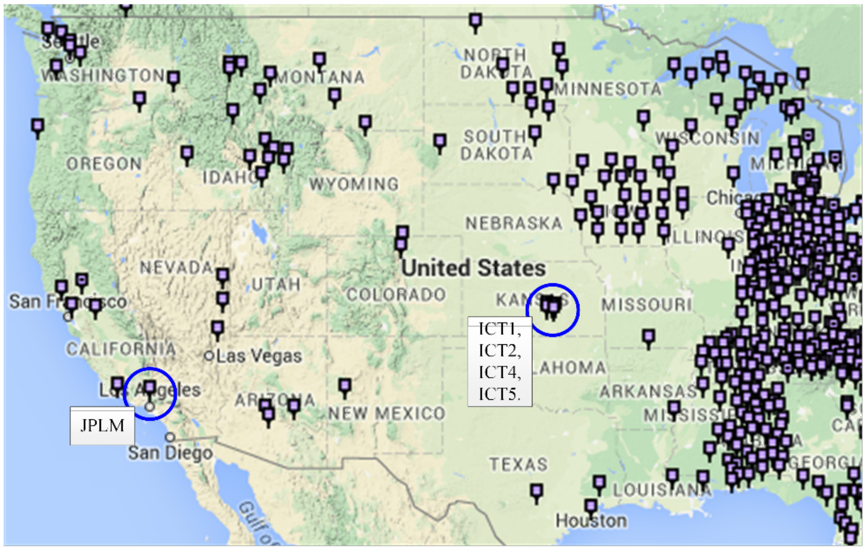


Figure II-4 Five CORS GNSS receivers with 1-second sampling rate used for the OOP evaluation.

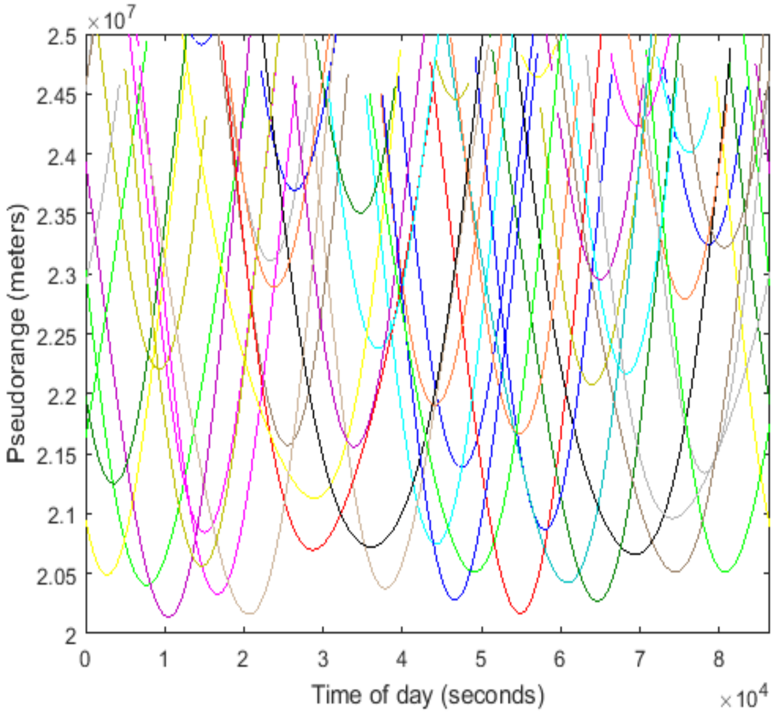


Figure II-5 L1 code pseudoranges of JPLM over day 145, 2015

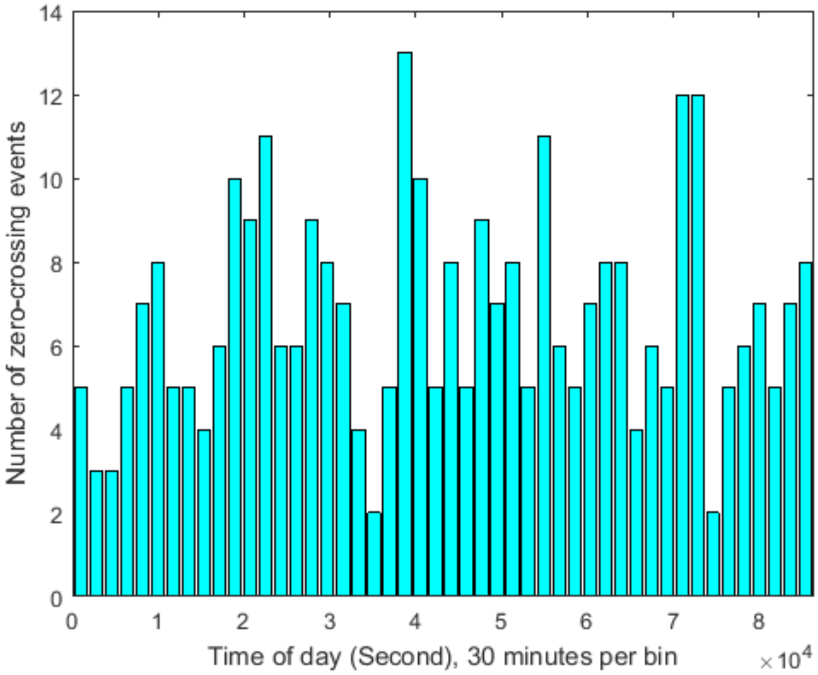


Figure II-6 Histogram of zero-crossing events measured by JPLM over day 145, 2015

Next, I examine zero-crossing events for the four closely spaced stations ICT1,2,4,5 and the result is plotted in Figure II-7, in which a zero crossing event is marked by a triangle oriented to south, east, west, and north, respectively. Due to their close proximity, these 4 stations are expected to have nearly identical times in recording their zero-crossing events. Most zero-crossing events fit the theory, as most marked shapes have all four events (e.g., the second lowest mark at the lowest left corner), but in several other marked locations, 1, 2 (first mark for index 350), or even 3 (the right most mark for index 350) marks were missing. The checking provides a simple glance at the consistency of different receivers at proximity of each other. The inconsistence could imply that they had bad measurements, bad logs, or malfunctions.

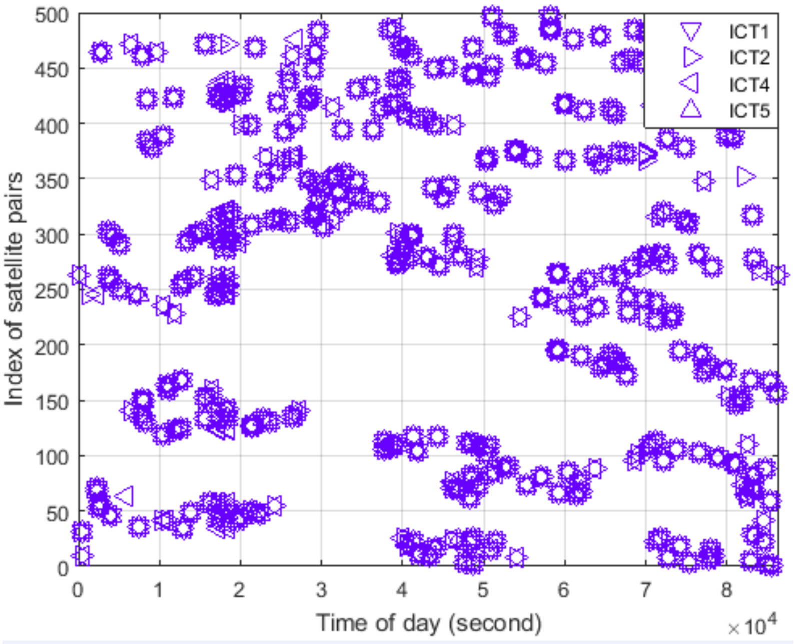


Figure II-7 Zero-crossing events of all satellite pairs measured by ICT1,2,4,5 over day 145, 2015

Next, I examine the paired zero-crossing events measured by ICT1 for 20 days in May and Jun, 2015, and the results for 5 selected days are plotted in Figure II-8. Using the 24 hours as a cycle, one can observe the shifting patterns for all pair indices. This is due to the fact that the periodicity of satellite orbit in the ECEF coordinate frame is on average 246 seconds less than one day period, from experiments done by A. Duncan and K. Larson [22].

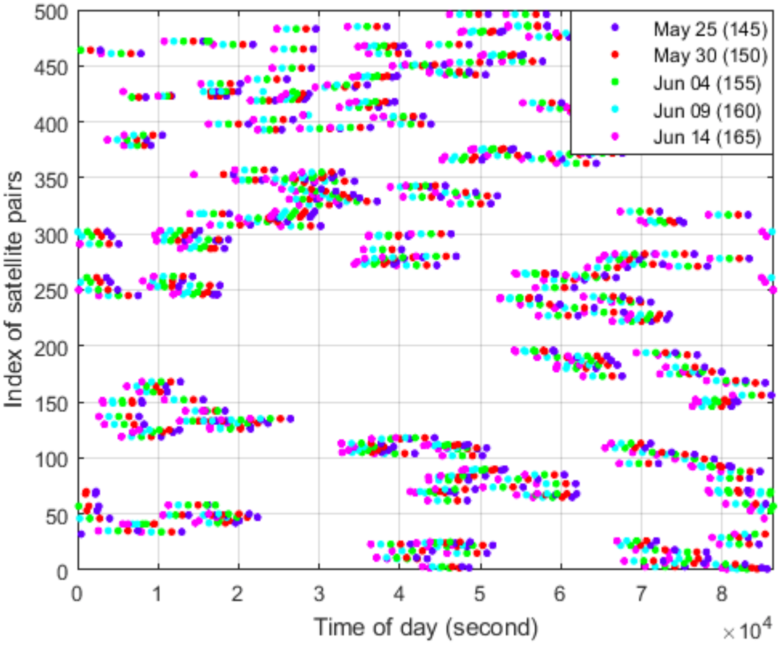
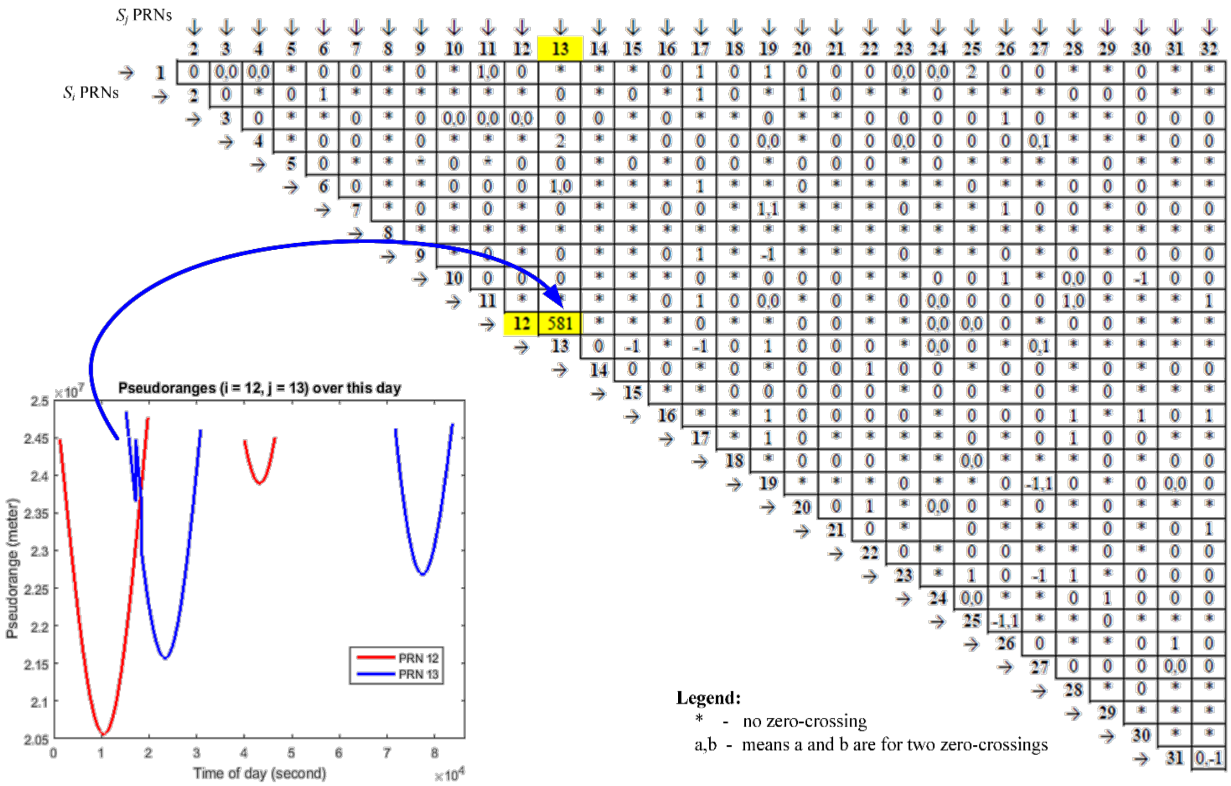


Figure II-8 Zero-crossing events of all satellite pairs measured by ICT1 on days 145, 150, 155, 160 and 160 in 2015

Last, but not least, we evaluate the performance of the proposed pairwise OOP model against the CORS data. Specifically, we used the ephemeris data to compute on the receiver's distance to the z-surface or d^z $(\overline{r^u})$, which was translated to the time when $d^z_{i,j}(\overline{r^u}) = 0$, and the time when zero-crossing event of $z_{i,j}$-surface was measured. We set the report rate of our computed zero-crossing events to one second, so that they are consistent with the 1-second report rate of CORS measurements. Key parameters used in our model include receiver u, satellite positions and semi-major axis a. The randomly chosen ICT1 as the receiver u has surveyed ECEF coordinate, as stated in its site log on the CORS website [23]. Satellite positions are computed with their ephemeris models given by the navigation files (RINEX) of ICT1.



S_j PRNs

S_i PRNs	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	0,0	0,0	*	0	0	*	0	*	1,0	0	*	*	*	0	1	0	1	0	0	0	0,0	0,0	2	0	0	*	*	0	*	*
2		0	*	0	1	*	*	*	*	*	*	0	*	0	*	1	0	*	1	0	*	*	0	*	*	*	0	0	0	*	*
3			0	*	*	0	*	0	0,0	0,0	0,0	0	0	*	0	*	*	0	*	*	0	0	0	0	1	0	*	*	0	0	0
4				*	0	0	*	0	*	*	*	2	*	*	0	0	0	0,0	*	0	0	0,0	0	0	0	0,1	*	*	*	0	0
5					0	*	*	*	0	*	0	0	*	0	*	0	0	*	0	0	0	*	0	*	*	*	*	0	*	*	*
6						0	*	*	0	0	0	1,0	*	*	*	1	*	*	0	*	*	*	*	0	*	*	0	0	0	*	*
7							*	0	*	0	*	0	*	*	0	0	*	1,1	*	*	*	0	*	*	1	0	0	*	0	0	0
8								*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
9									*	0	*	0	*	*	0	1	*	-1	*	*	*	0	*	*	0	*	0	*	0	0	0
10										0	0	0	*	*	*	0	*	0	0	0	*	*	0	0	1	*	0,0	0	-1	0	0
11											*	*	*	*	0	1	0	0,0	*	0	*	0	0,0	0	0	0	1,0	*	*	*	1
12												581	*	*	*	0	*	*	0	0	*	*	0,0	0,0	0	*	0	0	*	*	*
13													0	-1	*	-1	0	1	0	0	0	*	0,0	0	*	0,1	*	*	*	*	*
14														0	0	*	0	*	0	0	1	0	0	*	0	0	*	0	*	0	0
15															*	*	0	0	0	0	0	*	0	0	*	0	*	0	*	*	*
16																*	*	1	0	0	0	0	*	0	0	0	1	*	1	0	1
17																	*	1	0	*	*	*	*	0	*	0	1	0	0	*	*
18																		*	0	0	0	*	*	0,0	*	*	*	0	*	0	0
19																			*	*	*	0	*	*	0	-1,1	0	*	0	0,0	0
20																				0	1	*	0,0	0	*	0	*	0	*	*	*
21																					0	*		0	*	*	*	0	*	0	1
22																						0	*	0	0	*	*	0	*	0	0
23																							*	1	0	-1	1	*	0	0	0
24																								0,0	*	*	0	1	0	0	0
25																									-1,1	*	*	0	*	*	*
26																										0	*	*	0	1	0
27																											0	0	0	0,0	0
28																												*	0	*	*
29																													*	*	*
30																														*	*
31																															0,-1

Legend:

* - no zero-crossing

a,b - means a and b are for two zero-crossings

Figure II-9 Performance of the pairwise OOP model on ICT1 over day 145 in 2015

- **Matrix:** time difference (in second) of measured zero-crossings to modeled zero-crossings with respect to all satellite pairs.
- **Graph**: show the aberrant measurement of pseudorange from s_{13}.

For convenience, we used the ephemeris model at the beginning of the day, which is expected to have a few meters of error as it is expired. The semi-major axis a in OOP model contains ionosphere delay, troposphere delay and satellite clock bias. Among them, the satellite clock bias has the largest magnitude, which could be hundreds of microseconds, as one can find from the International GNSS Service or IGS orbit files in SP3 format [131]. In my experiment, this error is estimated using standard correction models with parameters in ephemeris [27]. Atmosphere delays can be readily included in the model using conventional models [27][37][38][39], but they were not included, representing a worse case assessment for the OOP model.

The performance of OOP model is measured based on the difference between the modeled zero-crossing times vs. that of the actual measurements. The comparison was made for the day 145 of 2015, and the result was shown in Figure II-9, in the format of a half matrix. Each row-column combination represents a satellite pair, and only half a matrix is required to represent all combinations. Within a day, 0, 1, or 2 zero-crossing events can be observed, and the differences between the modeled vs. measured times are entered. A "*" sign means that no zero-crossing was observed. For instance, no zero-crossing occurs to satellite 8. The number of zero-crossing pairs is not evenly distributed across the spectrum. For all observed cases except for two, the OOP has 1 second or less of error. The error of 581 and 2 seconds are respectively observed for the pair (s_{12}, s_{13}) and (s_4, s_{13}). The actual plots for s_{12} and s_{13} are plotted at left lower corner of Figure II-9, suggesting a spiking of the pseudorange for s_{13}. By adjusting the estimation of semi-major axis a, OOP can be readily applied to timing integrity check across different satellite channels, at the time resolution determined by the sampling and reporting rate of the reference receivers.

II.7 Conclusion

This chapter proposes the Order of Pseudoranges (OOP) as a system state for the signals of a set of satellites. We analyze the dynamics of z-surfaces generated from zero-crossing of the pseudoranges of satellite pairs. The resulting prediction model was tested using the 1-second pseudorange data from selected CORS sites.

Many interesting applications can be developed on the basis of the OOP model. For instance, the sweeping z-surfaces form natural temporal-spatial checkpoints for receivers to detect presence of aberrant signals. Receivers on the same z-surface have an overhead-free synchronization signal to the resolution of the error terms of the pseudorange equation. The number of z-surfaces will continue to grow with the deployment of new satellites.

From the perspective of distributed computing models, OOP offers a novel system state similar to the notion of virtual clock in Lamport's work, but it offers more than just temporal ordering. For instance, the z-surface for a satellite pair also offers a spatial order of receivers dynamically. The O-zone partitioned by z-surfaces is largely untouched due to space limit. How to explore these and many other untouched issues is critical to create the necessary computing abstractions, with provable properties, to integrate the powerful GNSS technology into the future generation of networking and distributed computing environments.

CHAPTER III Z-SURFACE & Z-CURVE – GNSS GLOBAL COMMON EVENTS[2]

III.1 Needs of Formalizing Z-Surface & Z-Curve

In this chapter, I aim to investigate the basic temporal and spatial properties of GNSS signals derived from their TDOA measures. Knowledge obtained from the modeling process will aid development of advanced applications. Knowing that positioning is the basis for remote sensing data analysis, the modeling techniques can be used for design of ground based monitoring grids to validate satellites' time and positions.

OOP aims to use the notion of *k-event*$^{(\mathcal{A},\,\mathcal{B})}$, when the TDOA of signals of $(\mathcal{A},\ \mathcal{B})$ is equal to k, to discretize states of GNSS signals at a position. The set of ground points that observe the same k-event form a *k-curve*$^{(\mathcal{A},\,\mathcal{B})}$. A k-event is called a *z-event* when k=0. In this chapter, I show that z/k-surface/curve can be characterized by several factors, such as satellite position, clock offset, and atmospheric delay, whose uncertainties are accounted for by existing models [32-41].

Geometric modeling of the TDOA can be solved using a hyperbolic framework, but most existing work is based on the planar environment. New techniques and their solvers are needed to characterize TDOA surfaces in 3D space. Key issues being addressed are summarized as follows:

1. The relationship among $(\mathcal{A},\,\mathcal{B})$ positions, clock errors, propagation errors, and their z-surface $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$.
2. The perimeter of a z-surface and its relation with signal coverage.

[2] Reprinted with permission from "Geometric Modeling of the Z-Surface and Z-Curve of GNSS Signals and Their Solution Techniques" by Guoyu Fu, Colton Riedel, Tyler Holmes, and Jyh-Charn Liu in *IEEE Transactions on Geoscience and Remote Sensing* 99 (2018): 1-12. Copyright 2019 by IEEE.

3. Geometrical interpretation of the *z*-surface, in terms of its eccentricity, foci, semi-major axis, opening directions.
4. The mathematical model of z-surface's ground intersection - z-curve $\mathbb{C}_z^{A,B}$.
5. Transformation of results from the mathematics-friendly Earth-Center Earth-Fixed (ECEF) coordinate frame to the user-friendly longitude-latitude coordinate frame.

Modeling of $\mathbb{S}_z^{A,B}$ is done by manipulating the pseudorange model while taking into account the signal propagation errors in pseudorange formation. The main challenge in modeling $\mathbb{S}_k^{A,B}$ is to find a valid range of k so that a feasible solution can be found. Modeling of $\mathbb{C}_z^{A,B}$ in the Cartesian form is not as difficult, but its parameterization is the well-known class of Quadric Surface Intersection Curve (QSIC) problems.

A high-level illustration of the *z*-surface and *z*-curve between ($\mathcal{A}$, $\mathcal{B}$) is given in Figure III-1Figure I-1, where the satellites' antennae emit GNSS signals in their TSV's at a time instant *t*. The primary factors which determine the propagation pathway of an RF signal are the satellite position, antenna direction, beam width, along several other parameters [29][30], which will be detailed in Chapter IV. The *z*-surface that represents the equal distance/pseudorange surface between ($\mathcal{A}$, $\mathcal{B}$) spans indefinitely in the 3D space. Its intersection with the Earth (surface) forms a *z*-curve, which is marked by the thick line segment on Figure III-1.

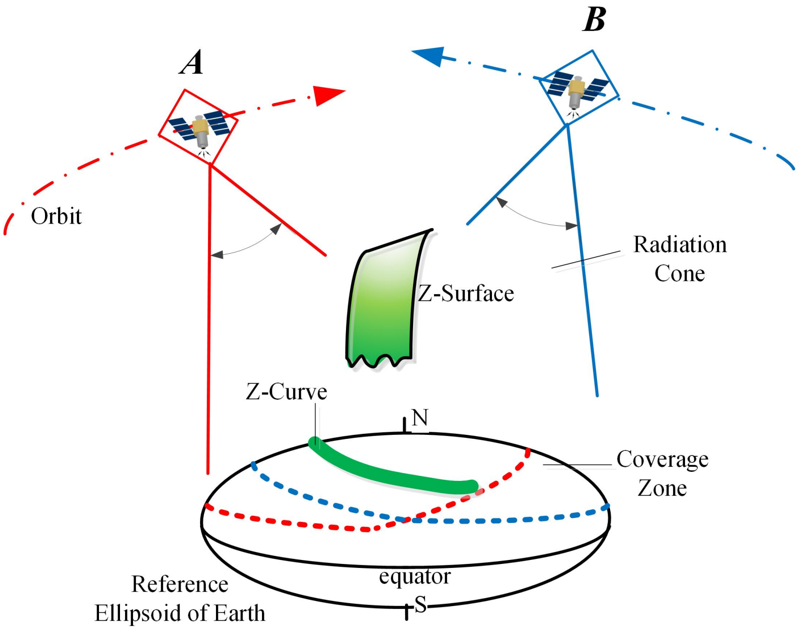


Figure III-1 Illustration of Geometric Components

For convenience, I assume the satellite position as a known value obtained from the broadcasted ephemeris model [27], because no closed form expression can accurately reflect satellite drift caused by outgassing, maneuvers, microgravity, solar winds, and other factors [28].

Let a 3D surface $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ denote the *z*-surface with respect to ($\mathcal{A}$, $\mathcal{B}$), where all points on $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ have zero difference in their pseudorange to ($\mathcal{A}$, $\mathcal{B}$). A *k*-surface, $\mathbb{S}_{\mathrm{k}}^{\mathrm{A,B}}$, is the generalization of a *z*-surface when the pseudorange difference is a constant value *k.* We will show that a *z*-surface (and *k*-surface) is also a quadratic shape, which is one sheet of two-sheeted hyperboloid when the pseudorange error terms are non-zero.

Given the positions of ($\mathcal{A}$, $\mathcal{B}$), ionospheric and tropospheric errors, and satellites' clock offsets, a *z*-surface in the unobstructed 3D space can be written in the general quadratic form:

$$\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}: \ \mathbf{x}^T M_Z \mathbf{x} + N_Z \mathbf{x} + L_Z = 0. \qquad \text{(III.1)}$$

Details will be explained in Theorems III.1 and III.2. The final constrained $\mathbb{S}_{\mathrm{k}}^{\mathrm{A,B}}$ has a similar form, as given in Theorem III.3. Parameters M_Z, N_Z, and L_Z describe the placement, orientation, eccentricity, and other characteristics of the hyperbolic sheet. If error terms of pseudoranges can be cancelled, the equation is reduced to a linear polynomial and the *z*-surface becomes a plane. To reflect real world conditions, the points of the *z*-surface receiving signals from ($\mathcal{A}$, $\mathcal{B}$) must be within $\mathbb{TSV}^{\mathrm{A,B}}$. Although this modeling technique can be applied to all elevations, for simplicity we only consider the altitudes below ("within") the troposphere. The flyover of $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ can be detected by a receiver Ω when a change of the order of pseudoranges occurs from $\rho_A > (<)\rho_B$ to $\rho_A < (>)\rho_B$.

A zero-crossing curve (*z*-curve) with respect to ($\mathcal{A}$, $\mathcal{B}$), denoted by $\mathbb{C}_{\mathrm{z}}^{\mathrm{A,B}}$, is the intersection of *z*-surface $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ and the Earth surface model $\mathbb{E}$ based-on its reference ellipsoid. This definition leads to the combination of (III.1) and (III.2) to form a system of two equations:

$$\mathbb{C}_{\mathrm{z}}^{\mathrm{A,B}} : \begin{cases} \mathbf{x}^T M_E \mathbf{x} + N_E \mathbf{x} + L_E = 0 \\ \mathbf{x}^T M_Z \mathbf{x} + N_Z \mathbf{x} + L_Z = 0 \end{cases}$$

Linear and quadratic constraints required to represent $\mathbb{TSV}^{\mathrm{A,B}}$ with respect to this *z*-surface will be discussed in Chapter IV. Although the *z*-curve offers more practical values for ground users than the *z*-surface does, it is not easy to generate points on a *z*-curve from the fourth-order (quartic) curve bounded by $\mathbb{TSV}^{\mathrm{A,B}}$. There is no simple way to find a parametric tracing of the QSIC [9-12], let alone find the constraints of the parameter. Therefore, in addition to the Cartesian representation of the *z*-curve, I also propose a parameterization method in Section III.5 to trace $\mathbb{C}_{\mathrm{z}}^{\mathrm{A,B}}$ and obtain the algebraic solution of each point $\mathbb{C}_{\mathrm{z}}^{\mathrm{A,B}}$.

III.2 Models of Earth Surface and Coordinate Systems

The real Earth's surface is the *topographic surface*, commonly indexed by its *geodetic latitude, longitude,* and *height* (LLH) [21], where the height is the vertical distance of a position on terrain above its sea level. Under the terrain, the sea level surface is a reference geoid surface derived from the distribution of micro-gravity [21]. Both the topographic and geoid surface models have no straightforward closed form, and are typically expressed in numerical formats such as Digital Elevation Models [20]. In this work we adopt the reference ellipsoid defined in WGS84 [18] [21] as the Earth's surface $\mathbb{E}$. The ellipsoid $\mathbb{E}$ can be expressed as:

$$\mathbb{E}:\ \frac{x^2}{a_E^2}+\frac{y^2}{a_E^2}+\frac{z^2}{b_E^2}=1. \qquad \text{(III.2)}$$

Here, a_E (6378137 meters) and b_E (≈6356752.314140 meters) are known constants which represent the equatorial radius and polar radius, respectively [18]; x, y, z are ECEF coordinates of a surface point [19]. An ellipsoid is a quadratic shape in 3D Euclidean space $\mathbb{R}^3$, which allows us to rewrite (III.2) in the quadratic form [8]:

$$\mathbb{E}:\ \mathbf{x}^T M_E \mathbf{x} + N_E \mathbf{x} + L_E = 0, \qquad \text{(III.3)}$$

where $\mathbf{x} = (x, y, z)^T \in \mathbb{R}^3$ is the coordinate of a receiver Ω, and

$$M_E = \begin{pmatrix} \frac{1}{a_E^2} & 0 & 0 \\ 0 & \frac{1}{a_E^2} & 0 \\ 0 & 0 & \frac{1}{b_E^2} \end{pmatrix},\quad N_E = (0,0,0),\quad L_E = -1.$$

The matrix representation of (III.1) will be extensively used for manipulations of linear and quadratic equations to derive geometric relationships among entities of interest in $\mathbb{R}^3$. I will use similar quadratic expressions for representations of the *z*-surface, and the symbol set

(M_x, N_x, L_x) will be used to represent three-dimensional coefficients of a quadratic polynomial expression of a system x, where "*x*=E" for the Earth, "*x*=V" for TSV, and "*x*=Z" for *z*-surface/*z*-curve.

The parametric Earth surface is denoted by the tuple (λ, φ), where longitude λ and geocentric latitude φ are spherical dimensions aligned with the ECEF frame. As illustrated in Figure III-2, λ is the angle between the prime meridian and a plane containing the North Pole, South Pole, and point of interest [18]. λ is positive when the point is east of the prime meridian (i.e. in the eastern hemisphere), and negative otherwise. φ is the angle between the equatorial plane and the line from Earth's center through the point [18]. It is positive when the point is on the northern hemisphere, and negative otherwise. (λ, φ)'s relationship with ECEF coordinates is bijective [18]:

$$\begin{cases} x = a_E \cos(\lambda)\cos(\varphi)\,, \\ y = a_E \sin(\lambda)\cos(\varphi)\,, \\ \quad z = b_E \sin(\varphi)\,, \end{cases} \tag{III.4}$$

where geocentric latitude's range is $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, and longitude's range is $\lambda \in [-\pi, \pi)$. It is important to note that φ is different from the more commonly used *geodetic latitude* ψ, as shown in Figure III-3. The bijective mapping between φ and ψ can be expressed as

$\psi = tan^{-1}(\frac{a_E^2}{b_E^2} \tan \varphi)$ [18].

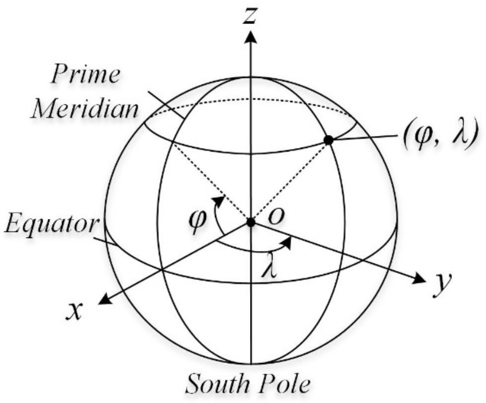


Figure III-2 The Earth coordinate systems

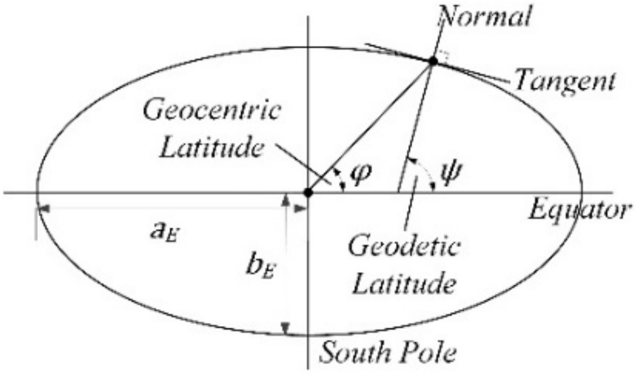


Figure III-3 The relationship between geocentric and geodetic latitudes.

The reference ellipsoid $\mathbb{E}$ approximates the real Earth surface and facilitates the development of a complete math system for the TSV. Readers are encouraged to replace the ellipsoid Earth model with other realistic models as needed for more precise analysis. The real Earth surface is the *topographic surface*, commonly indexed by its *geodetic latitude, longitude,* and *height* (LLH) [20], where the height is the vertical distance of a position on terrain above local sea level. Under the terrain is the sea level surface, given by a reference geoid surface derived from the distribution of micro-gravity [19]. Both topographic surface and geoid surface models have no closed form, and are typically expressed in numerical formats such as Digital Elevation Models (DEM) [20].

III.3 Cartesian Model of Z-Surface & Z-Curve

The *z*-surface $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ is the solution space of pseudorange difference $\Delta^{A,B}(\mathbf{x}) = 0, \forall\mathbf{x} \in \mathbb{TSV}^{\mathrm{A,B}}$. Let $\mathfrak{S}^{A,B}$ denote this solution space but without the condition "$\forall\mathbf{x} \in \mathbb{TSV}^{\mathrm{A,B}}$". We can derive expression for $\mathfrak{S}^{A,B}$ first, and then add the constraint of $\forall\mathbf{x} \in \mathbb{TSV}^{\mathrm{A,B}}$ to the system of expressions.

$\mathfrak{S}^{A,B}$ is a set of points in space at which a GNSS receiver measures the same pseudoranges from ($\mathcal{A}$, $\mathcal{B}$) at time t. The pseudorange model [25][26][28] is given as the sum of geometric distance and error terms:

$$\rho = \|\mathbf{x}^S - \mathbf{x}\| + v\delta_{uc}(\mathbf{x}) + vE1(\mathbf{x}) + vE2(\mathbf{x}), \tag{III.5}$$

$$E1(\mathbf{x}) = \delta_{io}(\mathbf{x}) + \delta_{tr}(\mathbf{x}) - \delta_{sc}(\mathbf{x}), \tag{III.6}$$

where v is the speed of light in vacuum, δ_{uc} the receiver clock offset, $E1$ represents error terms of a GNSS system (ionospheric delay δ_{io}, tropospheric delay δ_{tr}, satellite clock error δ_{sc}), *E2* denotes man-made sources like spoofing, multipath [30][31]. By substituting (5) in $\Delta^{A,B}(\mathbf{x}) = 0$, one can readily obtain:

$$\Delta^{A,B}(\mathbf{x}) = \|\mathbf{x}^B - \mathbf{x}\| - \|\mathbf{x}^A - \mathbf{x}\| + v\big(E1^B(\mathbf{x}) - E1^A(\mathbf{x})\big) + v\big(E2^B(\mathbf{x}) - E2^A(\mathbf{x})\big). \tag{III.7}$$

When $E2 = 0$, $v(E2^B - E2^A) = 0$ in (7). As such, $\mathfrak{S}^{A,B}$ represents a set of points satisfying

$$\begin{aligned}\sqrt{(x^B - x)^2 + (y^B - y)^2 + (z^B - z)^2} - \sqrt{(x^A - x)^2 + (y^A - y)^2 + (z^A - z)^2} \\ + v(E1^B - E1^A) = 0,\end{aligned} \tag{III.8}$$

where the last linear term is a constant by assumption. For convenience, we let $a = \frac{v}{2}(E1^A - E1^B) = \frac{v}{2}[(\delta^A_{io} - \delta^B_{io}) + (\delta^A_{tr} - \delta^B_{tr}) - (\delta^A_{sc} - \delta^B_{sc})]$. a is assumed to be a constant for the areas of concern, because δ_{uc}'s from A and B's pseudoranges are same and cancelled out, while δ_{io}, δ_{tr} and δ_{sc} are assumed to be constant at a given time instance [25][26]. It will become clear shortly that a is essentially the *semi-major axis* of the solution space $\mathcal{S}$ to be solved in Theorem III.1.

In the following discussion, a necessary and sufficient condition for $\mathfrak{S}^{A,B}$ being a non-empty set is given in Lemma III.1. The geometric meaning and property of $\mathfrak{S}^{A,B}$ is discussed in Theorem III.1. A compact Cartesian system of $\mathfrak{S}^{A,B}$ is presented in Lemma III.2, by rewriting (III.8). Finally, the Cartesian expression for $\mathbb{S}_z^{A,B}$ is provided in Theorem III.2 by incorporating the system for $\mathbb{TSV}^{A,B}$ into the system of $\mathfrak{S}^{A,B}$.

Lemma III.1 *$\mathfrak{S}^{A,B}$ is a non-empty set if and only if $-c < a < c$, where $c = \frac{1}{2}\|\mathbf{x}^A - \mathbf{x}^B\|$ is a half of the geometric distance between $\mathcal{A}$* and *$\mathcal{B}$.*

Proof: When $\mathfrak{S}^{A,B}$ is a non-empty set, there exists at least an arbitrary point q which forms a triangle with ($\mathcal{A}$, $\mathcal{B}$) in the 3D space $\mathbb{R}^3$, so that the following inequality must be satisfied: $|\|\mathbf{x}^B - \mathbf{x}^q\| - \|\mathbf{x}^A - \mathbf{x}^q\|| < \|\mathbf{x}^A - \mathbf{x}^B\|$. By substituting (III.8) into the inequality expression, we get $-c < a < c$. Conversely, when $-c < a < c$, $\mathfrak{S}^{A,B}$ is a non-empty set, because $|2a| < 2c$ implies that there are points which can form a triangle with ($\mathcal{A}$, $\mathcal{B}$). That is, some of these points in $\mathbb{R}^3$ satisfy (III.8). ■

As will be shown in Theorem III.1, $\mathfrak{S}^{A,B}$ is a hyperboloid at most times, and a and c are two factors of its eccentricity e. Empirical results show that a can be either negative or positive, and that a's magnitude is typically less than c. That being said, $\mathfrak{S}^{A,B}$ exists and is usually not empty based on Lemma III.1. a is primarily determined by the difference of satellites' clock offset $\delta_{sc}^{A} - \delta_{sc}^{B}$, instead of atmospheric delays, because δ_{sc} can be as large as 1 millisecond, or 300 kilometers equivalently, while δ_{io} and δ_{tr} are on the magnitude of tens of nanoseconds [26]. Even though these errors can be well compensated for positioning, their physical values before compensation will affect the curvature of z-surface.

Theorem III.1 When $E1^{B} = E1^{A}$, *$\mathfrak{S}^{A,B}$ is an equal distance plane between ($\mathcal{A}$, $\mathcal{B}$). Otherwise, when $E1^{B} \neq E1^{A}$, $\mathfrak{S}^{A,B}$ is one half sheet of a two-sheeted hyperboloid, whose foci are $\mathcal{A}$, $\mathcal{B}$, and the semi-major axis is a.*

The geometric meaning of Theorem III.1 is illustrated in Figure III-4. The solution space $\mathfrak{S}^{A,B}$ can be one of the three sub-shapes $H^{A,B}$, *wing-B (W^{B}*), and *wing-A* (W^{A}) of a hyperboloid shape. Let q denote an arbitrary point on $\mathfrak{S}^{A,B}$, W^{B} represents the case when $a = \frac{1}{2}(d(B,q) - d(A,q))$ is less than zero, $H^{A,B}$ the case when $a = 0$, and W^{A} the case when $a > 0$, where the d function refers to the geometric distance between two points.

Proof: To solve $\mathfrak{S}^{A,B}$, we rewrite (8) by replacing $v(E1^{B} - E1^{A})$ with $2a$:

$$\sqrt{(x^{B} - x)^2 + (y^{B} - y)^2 + (z^{B} - z)^2} - \sqrt{(x^{A} - x)^2 + (y^{A} - y)^2 + (z^{A} - z)^2} = 2a, \quad (9)$$

where the two square root terms on the left-hand side (LHS) represent the geometric distances from an arbitrary point q to ($\mathcal{A}$, $\mathcal{B}$), or $d(A,q)$ and $d(B,q)$, respectively. The "$2a$" on the right-hand side (RHS) is the difference from $d(B,q)$ to $d(A,q)$.

The special case of $\mathfrak{S}^{A,B}$ with $a = 0$ is represented by $H^{A,B}$, in Figure III-4, and (III.8) can be rewritten in the general form

$$C_z x + D_z y + E_z z + F_z = 0, \quad \text{(III.9)}$$

where $C_z = 2(x^A - x^B), D_z = 2(y^A - y^B),\ E_z = 2(z^A - z^B)$, and $F_z = \mathbf{x}^{B^T}\mathbf{x}^B - \mathbf{x}^{A^T}\mathbf{x}^A$. The first three coefficients are elements of the normal vector of $H^{A,B}$, i.e. $(C_z, D_z, E_z)^T$, which is exactly in the direction of a unit vector $\bar{n}$ from $\mathcal{B}$ to $\mathcal{A}$:

$$\bar{n} = \left(\frac{x^A - x^B}{2c}, \frac{y^A - y^B}{2c}, \frac{z^A - z^B}{2c}\right)^{\mathrm{T}},$$

where c refers to half of distance between $\mathcal{A}$ and $\mathcal{B}$ as defined in Lemma III.1. This implies that $H^{A,B}$ is always perpendicular to the line connecting A and B. Moreover, by substituting $o' = \frac{1}{2}(\mathbf{x}^A + \mathbf{x}^B)$, the midpoint of A and B, into (III.9), one finds that $H^{A,B}$ always passes through o'. Therefore, $H^{A,B}$ is a special plane and every point on $H^{A,B}$ has equidistant to A and B.

In case of $a \neq 0$, let $\mathfrak{S}^{A,B}(a \neq 0)$ denote a set of points satisfying (III.8). Formally, for all points $q \in \mathfrak{S}^{A,B}(a \neq 0),\ d(B,q) - d(A,q) = 2a \neq 0$. I could not locate any 3D shape definitions in literature to represent $\mathfrak{S}^{A,B}(a \neq 0)$, therefore we present a simple technique to construct $\mathfrak{S}^{A,B}(a \neq 0)$ from the collection of 2D planes, each of which is a slice intersecting the line segment AB, with all inclination angles $\theta \in [0, \pi)$. Let $\mathbb{R}^2(\theta)$ denote each of these 2D section planes. The trace of points on the plane $\mathbb{R}^2(\theta)$ have the same distance difference $2a$ to two foci, by (III.8). Therefore, the portion of $\mathfrak{S}^{A,B}(a \neq 0)$ on $\mathbb{R}^2(\theta)$ is a hyperbola with ($\mathcal{A}$, $\mathcal{B}$) as the foci and a as the semi-major axis, by the definition of a hyperbola [118]. One such hyperbolic curve is plotted in Figure III-4. Note also that on $\mathbb{R}^2(\theta)$, the geometric meaning of (III.8) is slightly different from the definition of hyperbola, in that (III.8) is not expressed in the absolute value of $d(B,q) - d(A,q)$. As a result, $\mathfrak{S}^{A,B}(a \neq 0)$ on $\mathbb{R}^2(\theta)$ refers to only one half (wing) of a hyperbola. By iterating over all $\theta \in [0, \pi)$, the union set of all hyperbola wings form a single sheet of a two-sheeted hyperboloid of revolution [118], whose characteristics are determined by $\mathbf{x}^A$, $\mathbf{x}^B$, c and a. Specifically, when $a > 0$, or $d(B,q) > d(A,q)$, $\mathfrak{S}^{A,B}(a \neq 0)$ is W^A. Otherwise, when $d(A,q) > d(B,q)$, $\mathfrak{S}^{A,B}(a \neq 0)$ is wing-*B*, W^B. ■

From Theorem III.1, one can write the canonical form of $\mathfrak{S}^{A,B}(a \neq 0)$ by placing it on a local coordinate $L^{A,B}$, which can be constructed by letting o' be $L^{A,B}$'s origin, $\bar{n}$ the y'-axis of $L^{A,B}$, and any right-handed orthogonal vectors on $H^{A,B}$ the (x'-, z'-) axes of $L^{A,B}$. In $L^{A,B}$, coordinates of ($\mathcal{A}$, $\mathcal{B}$) are respectively $\mathbf{x}^{A\prime} = (0, c, 0)$ and $\mathbf{x}^{B\prime} = (0, -c, 0)$. And $\mathfrak{S}^{A,B}(a \neq 0)$ in $L^{A,B}$ has the form

$$\frac{x\prime^2}{c^2-a^2} - \frac{y'^2}{a^2} + \frac{z'^2}{c^2-a^2} = -1, \text{where } ay' > 0, \tag{III.10}$$

where the equation describes the collection of W^A and W^B of the hyperboloid, and the inequality constraint the solution points on only one of the two wings. Since Theorem III.1 implies $\mathfrak{S}^{A,B}$ is a quadratic shape of hyperboloid, $\mathfrak{S}^{A,B}$ can be represented in a general quadratic form in both $a = 0$ and $a \neq 0$ cases. Lemma III.2 below gives a compact representation for $\mathfrak{S}^{A,B}$.

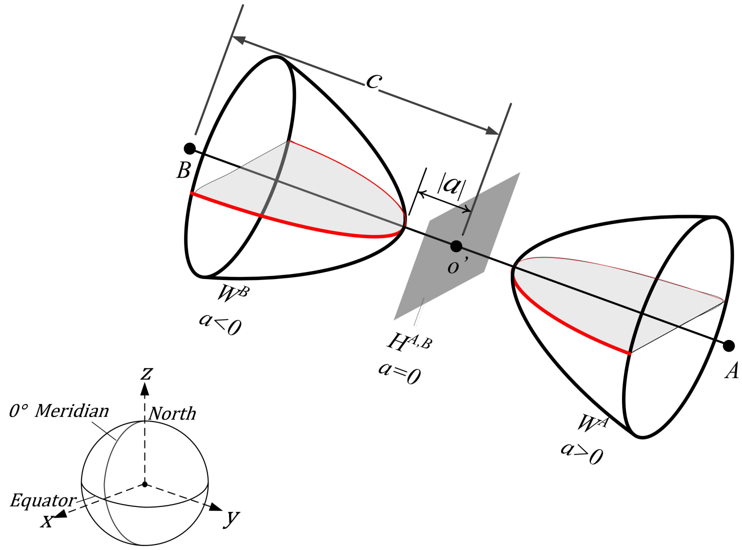


Figure III-4 Illustration of $\mathfrak{S}^{A,B}$ on the ECEF frame.

Lemma III.2 *A quadric matrix form of (III.8) can be expressed as*

$$\begin{cases} \mathbf{x}^T M_Z \mathbf{x} + N_Z \mathbf{x} + L_Z = 0, \\ a(C_Z x + D_Z y + E_Z z + F_Z) \geq 0, \end{cases}$$

where $M_Z = \{m_{i,j}\}, i, j = 1{,}2{,}3$, $N_Z = \{n_i\}, i = 1{,}2{,}3$, $L_Z \in \mathbb{R}$.

The variables are defined as follows:

$$M_Z = \begin{pmatrix} C_Z^2 - 4a^2 & C_Z D_Z & C_Z E_Z \\ C_Z D_Z & D_Z^2 - 4a^2 & D_Z E_Z \\ C_Z E_Z & D_Z E_Z & E_Z^2 - 4a^2 \end{pmatrix},$$

$$N_Z = \begin{pmatrix} 2G_Z C_Z + 8a^2 x^A \\ 2G_Z D_Z + 8a^2 y^A \\ 2G_Z E_Z + 8a^2 z^A \end{pmatrix},$$

$L_Z = G_Z^2 - 4a^2 \mathbf{x}^{A^T} \mathbf{x}^A$, $G_Z = F_Z - 4a^2$.

Proof: Rewrite the following equation into a quadric matrix form:

$$\sqrt{(x^B - x)^2 + (y^B - y)^2 + (z^B - z)^2} - \sqrt{(x^A - x)^2 + (y^A - y)^2 + (z^A - z)^2} = 2a.$$

Step 1: Move one of the square root (e.g. the distance to satellite A) to the RHS, and square both sides. We get:

$$x^2 + y^2 + z^2 - 2x^B x - 2y^B y - 2z^B z + {x^B}^2 + {y^B}^2 + {z^B}^2 =$$

$$4a^2 + 4a\sqrt{(x^A - x)^2 + (y^A - y)^2 + (z^A - z)^2} + x^2 + y^2 + z^2 - 2x^A x - 2y^A y - 2z^A z +$$

$${x^A}^2 + {y^A}^2 + {z^A}^2,$$

Step 2: Leave the term of square root at the RHS, and substitute ${x^A}^2 + {y^A}^2 + {z^A}^2$ and ${x^B}^2 + {y^B}^2 + {z^B}^2$ with $\mathbf{x}^{A^T}\mathbf{x}^A$ and $\mathbf{x}^{B^T}\mathbf{x}^B$, respectively. We get:

$$G + Cx + Dy + Ez = 4a\sqrt{(x_A - x)^2 + (y_A - y)^2 + (z_A - z)^2},$$

where $G, C, D,$ and E are given as:

$$G = \mathbf{x}^{B^T}\mathbf{x}^B - \mathbf{x}^{A^T}\mathbf{x}^A - 4a^2,\ C = 2(x_A - x_B),$$

$$D = 2(y_A - y_B),\ E = 2(z_A - z_B).$$

Step 3: Square both sides and after some reorganization of the terms, we get:

$$m_{1,1}x^2 + m_{2,2}y^2 + m_{3,3}z^2 + 2m_{1,2}xy + 2m_{1,3}xz + 2m_{2,3}yz + k_1x + k_2y + k_3z + L = 0,$$

where coefficients are given as below:

$$m_{1,1} = C^2 - 16a^2,\ m_{2,2} = D^2 - 16a^2,$$

$$m_{3,3} = E^2 - 16a^2, m_{1,2} = CD,\ m_{2,3} = DE,\ m_{1,3} = CE,$$

$$k_1 = 2GC + 32a^2x_A,\ k_2 = 2GD + 32a^2y_A,\ k_3 = 2GE + 32a^2z_A,$$

$$L = G^2 - 16a^2\mathbf{x}^{A^T}\mathbf{x}^A.$$

Its matrix form is:

$$\mathbf{x}^T M\mathbf{x} + K\mathbf{x} + L = 0,$$

where $M = \{m_{i,j}\}, i, j \in \{1,2,3\}$, and $K = \{k_i\}$, $i \in \{1,2,3\}$. It is important to note that only half of the solutions should be used, as this equation also covers the situation of $\|\mathbf{x}^B - \mathbf{x}\| - \|\mathbf{x}^A - \mathbf{x}\| = -2a$. ■

Theorem III.2 *The z-surface of* $(\mathcal{A}, \mathcal{B})$, $\mathbb{S}_Z^{A,B}$, *in the implicit Cartesian expression is*

$$\mathbb{S}_Z^{A,B}: \begin{cases} \mathbf{x}^T M_E\mathbf{x} + N_E\mathbf{x} + L_E \geq 0, & \text{I} \\ \mathbf{x}^T M_Z\mathbf{x} + N_Z\mathbf{x} + L_Z = 0, & \text{II} \\ a(C_Zx + D_Zy + E_Zz + F_Z) \geq 0, & \text{III} \\ \mathbf{x}^T M_V^A\mathbf{x} + N_V^A\mathbf{x} + L_V^A \leq 0, & \\ C_V^Ax + D_V^Ay + E_V^Az + F_V^A \geq 0, & \\ \boldsymbol{x}^T M_G^A\boldsymbol{x} + N_G^A\boldsymbol{x} + L_G^A \leq 0, & \\ \mathbf{x}^T M_V^B\mathbf{x} + N_V^B\mathbf{x} + L_V^B \leq 0, & \\ C_V^Bx + D_V^By + E_V^Bz + F_V^B \geq 0, & \\ \boldsymbol{x}^T M_G^B\boldsymbol{x} + N_G^B\boldsymbol{x} + L_G^B \leq 0, & \end{cases} \qquad (\text{III}.11)$$

where the 2nd (II) and 3rd (III) expressions represent the unconstrained *z*-surface $\mathfrak{S}^{A,B}$ with respect to ($\mathcal{A}$, $\mathcal{B}$) as described in Lemma 3, the rest inequalities describe $\mathbb{TSV}^{\mathrm{A,B}}$ described in Chapter IV.

III.4 Extension to K-Surface & K-Curve

The *k*-surface $\mathbb{S}_{\mathrm{k}}^{\mathrm{A,B}}$ is a close relative of the *z*-surface to represent the surface where points have same pseudorange difference k to ($\mathcal{A}$, $\mathcal{B}$). Let a_k denote a constant value of $a_k = \frac{c}{2}(E1_t^A - E1_t^B) + \frac{k}{2}$. Equation $\Delta^{A,B}(\mathbf{x}) = k$ is rewritten as:

$$\sqrt{(x^B - x)^2 + (y^B - y)^2 + (z^B - z)^2} - \sqrt{(x^A - x)^2 + (y^A - y)^2 + (z^A - z)^2} = 2a_k. \tag{III.12}$$

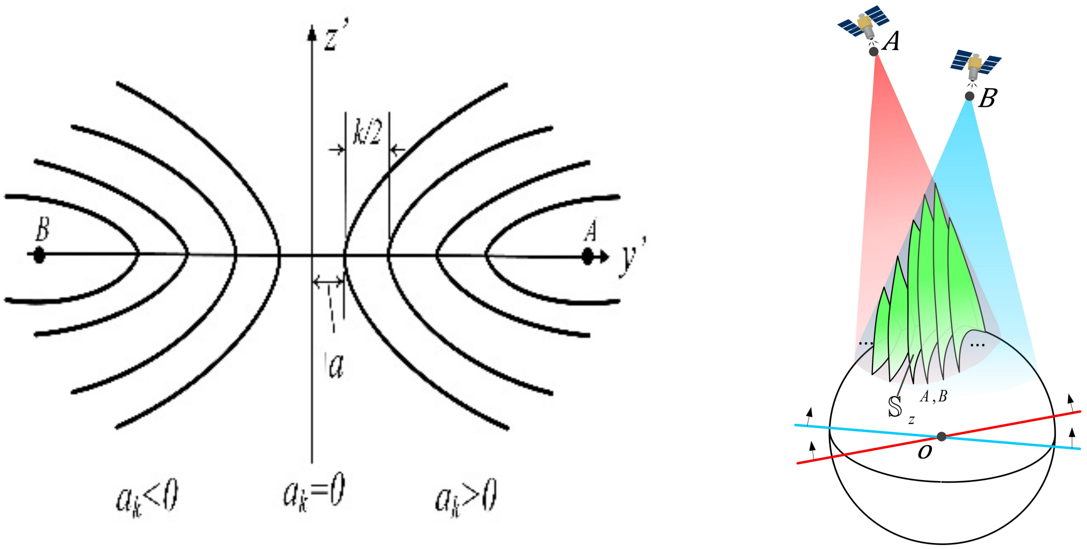
Lemma III.3: *The solution space of (III.12) is a non-empty set if and only if* $-2c - 2a < k < 2c - 2a$.

Proof: Omitted here, since it is similar to that of Lemma III.1. ■

Theorem III.3 *The k-surface of* ($\mathcal{A}$, $\mathcal{B}$), $\mathbb{S}_{\mathrm{k}}^{\mathrm{A,B}}$, *in the implicit Cartesian expression is identical to that of* $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ *in Theorem III.2, except that the coefficient a in expression III is replaced by* a_k.

By substituting different values of k, one can obtain *k*-surfaces, as illustrated in Figure III-5. Intuitively, the distance between the two wings of the hyperboloid increases with the *k* value, while the openness of the wing is narrowed.

The system of equations for $\mathbb{C}_{\mathrm{k}}^{\mathrm{A,B}}$ is derived from Theorem III.2 by replacing a by a_k, and "≤"with"=" in the first expression to constrain the points in $\mathbb{C}_{\mathrm{k}}^{\mathrm{A,B}}$ to the Earth's surface.



(a) on the local $y'z'$ section (b) on the global ECEF frame

Figure III-5 The k-surfaces with different k values.

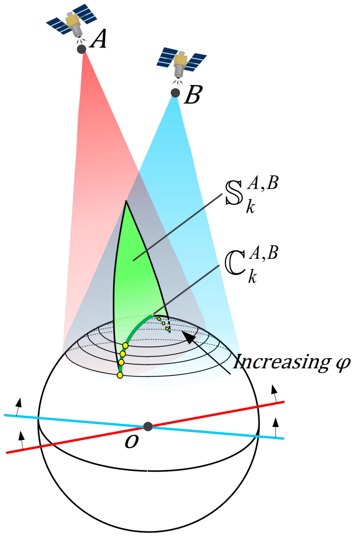


Figure III-6 Parameterization schemes for coverage zone and z-curve.

Theorem III.4 *The implicit expression of a z(k)-curve for satellites A and B and about the reference ellipsoid defined in WGS84, is given by replacing expressions I, III in Theorem III.2 as follows:*

I → $$\mathbf{x}^T M_E \mathbf{x} + N_E \mathbf{x} + L_E = 0,$$

III→ $$\alpha(C_Z x + D_Z y + E_Z z + F_Z) \geq 0,$$

where $\alpha = a$ *for a z-curve, and* $\alpha = a_k$ *for a k-curve.*

III.5 Parameterization of Z-Curve

In many applications like computer visualization and mapping, Cartesian models are inconvenient because they require a full enumeration for all points in the 3D space. A *parameterization* scheme is discussed here to trace the *z*(*k*)-curve based on only one parameter, i.e. the geocentric latitude φ. The longitude λ is positive when the point is east of the prime meridian, and negative otherwise. The geocentric latitude φ is positive when the point is in the northern hemisphere, and negative otherwise. (λ, φ)'s relationship with ECEF coordinates is bijective and given by [19]:

$$\begin{cases} x = a_E \cos(\lambda)\cos(\varphi)\,, \\ y = a_E \sin(\lambda)\cos(\varphi)\,, \\ \quad z = b_E \sin(\varphi)\,, \end{cases} \tag{III.13}$$

where $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, $\lambda \in [-\pi, \pi)$. It is important to note that φ is different from the more commonly used *geodetic latitude* ψ. Their relation can be expressed as $\psi = tan^{-1}(\frac{a_E^2}{b_E^2}\tan\varphi)$ [132].

The proposed representations with (λ, φ) are illustrated in Figure III-6. The *z*-curve $\mathbb{C}_{\mathrm{k}}^{\mathrm{A,B}}$ is discretized into dots (in yellow), one or two of which are located at the same φ-circle but with different longitudes. Therefore, both shapes can be described by mathematical ranges: let $\mathbb{R}_\varphi$ denote the valid latitude range, and for each $\varphi \in \mathbb{R}_\varphi$, and let $\mathbb{R}_\lambda(\varphi)$ denote the longitude range on the φ-circle for the coverage zone, or the longitude value(s) of point(s) on the z-curve.

Being a QSIC, $\mathbb{C}_{\mathrm{k}}^{\mathrm{A,B}}$ is an ellipsoid-hyperboloid shape. Neither an ellipsoid nor a hyperboloid is a ruled surface (i.e. a surface that can be described as a cluster of lines). It thus remains a hard problem to efficiently parameterize the ellipsoid-hyperboloid intersection (EHI) algebraically. A well known method proposed by Levin *et al* [2] can search for a ruled surface in the *pencil* [11] (i.e. the cluster of shapes crossing the intersection of the two shapes) of two general quadric shapes, and thus reduce the final polynomial from the 4th to the 2nd order. However, it has non-trivial computations unrelated to the specific EHI problem. In our work, we directly substitute the parametric Earth (III.13) into the Cartesian expression of the hyperboloid, i.e. the equation II in (III.11). The derivation is detailed in Appendix 1. The result will be a 4th order polynomial equation of longitude value λ, given a known geocentric latitude φ. Every time a pair of (λ, φ) is solved, it must be converted to a Cartesian coordinate to be tested against the remainder of the constraints in (III.11). φ becomes the tracing parameter: for each geocentric latitude $\varphi \in \mathbb{R}_{\varphi}$, the real roots of a 4th order equation, if also satisfying the rest constraints in (III.11), can be the longitudinal value(s) λ, which label the point(s) of the *z*-curve on this φ-circle.

III.6 Experimental Validation

A C++ based software tool was developed to validate the Cartesian models and their parameterization methods using the data from almost 4000 GNSS reference stations in the public IGS and CORS databases [35][36]. The tool implements construction of Cartesian models, parameterization functions, polynomial solvers, time and coordinate converters, parsers for RINEX observation and navigation files, and the sp3 file, and utility functions. *OpenGL* and *CGAL* libraries [23][121] were used for 3D visualization and interactive adjustment. A 2D display webpage was implemented using python and *Google Maps* to display the overlays of reference stations and *z*-curves. Other numerical experiments are carried out in *Matlab*.

Form Theorem III.3, z-curves between satellite pairs can be generated for their expected ground (Cartesian) positions at different times. After parametrization, they are compared against the 1-Hz measurements made at the 190 IGS and 756 CORS networks of GPS/GNSS receivers. Snapshots of the z-curves for a GPS satellite pair (6, 30) at 30-second intervals (denoted in Time of Week, or TOW in second) are plotted in the left diagram of Figure III-7. CORS and IGS stations are marked as circles initially. They are replaced by crosses when experiencing zero-crossings of pseudoranges from satellites (6, 30), i.e. the transition from $\rho^6 - \rho^{30} > 0$ to $\rho^6 - \rho^{30} < 0$, vice versa. Nearly all stations detect zero-crossings at the times predicted by the proposed z-curve model. Some stations during this time are not marked as crosses due to lack of measurements. Among more than 400 curves, this series of z-curves ($\mathbb{C}_z^{6,30}$) were chosen for illustration because they passed the greatest number of reference stations that day.

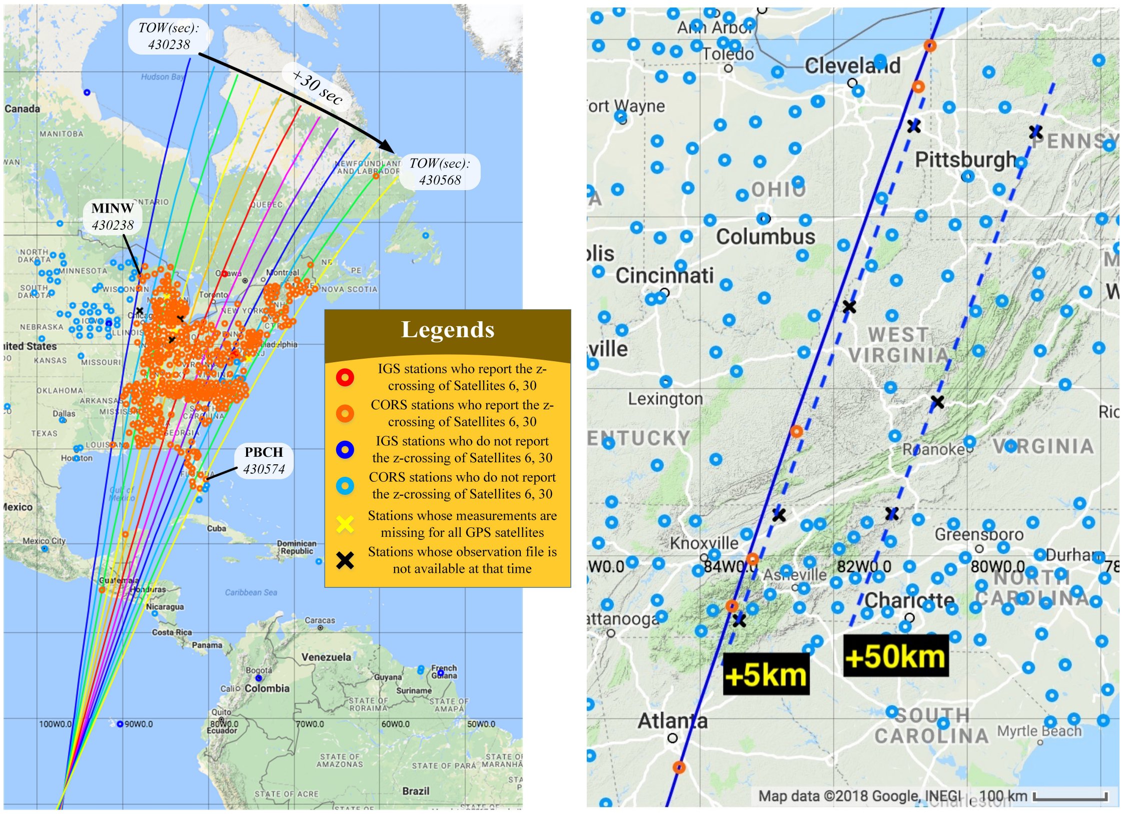

TOW(sec): 430238
+30 sec
TOW(sec): 430568
MINW 430238
PBCH 430574
Legends
IGS stations who report the z-crossing of Satellites 6, 30
CORS stations who report the z-crossing of Satellites 6, 30
IGS stations who do not report the z-crossing of Satellites 6, 30
CORS stations who do not report the z-crossing of Satellites 6, 30
Stations whose measurements are missing for all GPS satellites
Stations whose observation file is not available at that time
+5km
+50km
Map data ©2018 Google, INEGI
100 km

Figure III-7 Snapshots of z-curves vs. detections of IGS and CORS.

- **Left**: real z-curves and detected stations in 330 seconds from TOW 430238 to 430568 seconds (12 snapshots are shown)
- **Right**: a z-curve in solid line and two simulated z-curves in dash lines (detected stations are marked in stars), at TOW 430388 second
- **Note**: the cluster of z-curves are $\mathbb{C}_z^{6,30}$, Day 110, 2017. Stations are marked as circles, and become crosses when detecting z-curve.

Reporting a *z*-curve crossing at an unexpected time by a group of receivers indicates anomaly with respect to a common event. A potential application of the *z*-curve model is therefore detection of anomalous satellite trajectories, spoofing, or atmosphere properties, especially the ionosphere, which can be disturbed by various causes [133]. We use the right diagram of Figure III-7 to show the simulation result. Given a *z*-curve of $\mathbb{C}_z^{6,30}$ at TOW 430388, it is expected that 6 stations (crosses) report its crossing within ± 1 second, shown as the solid line. If ρ^{30} is increased by 5,000 and 50,000 meters, the stations marked as stars instead would observe $\mathbb{C}_{\mathrm{z}}^{6,30}$. Since the same *z*-curve should reappear periodically, this anomaly checking logic can be scheduled.

Another potential application is the collaborative survey of GNSS constellation orbits and geodetic models. Any discrepancy between the predicted and measured zero-crossings is the result of imperfections of the Earth's surface model, atmospheric models, orbital models, or signal models. The 1-Hz reporting rate of receivers used in this dissertation is not fine enough to reveal any of these potential imperfections. However, when GNSS receivers report pseudoranges every millisecond, any error of zero-crossing predictions would be more detectable, thereby revealing any potential imperfections in geodetic or navigation models.

Multiple *z*-curves may intersect simultaneously. In Figure III-8, I randomly select three satellites (PRNs 2, 3, 30). Eight instances, on 15-minute intervals, of *z*-curves $\mathbb{C}_{\mathrm{z}}^{2,3}$, $\mathbb{C}_{\mathrm{z}}^{2,30}$, $\mathbb{C}_{\mathrm{z}}^{3,30}$ are plotted. Initially (TOW 345420 second) they have no intersection, but appear to be converging at some point outside of their co-coverage zone. At the next instance, the three curves intersect at a point. The three *z*-curves intersect at a different circled point at each subsequent observation. Consider three satellites with PRNs i, j, k $(i < j < k)$ which form three intersecting *z*-curves $\mathbb{C}_{\mathrm{z}}^{\mathrm{i,j}}$, $\mathbb{C}_{\mathrm{z}}^{\mathrm{i,k}}$, $\mathbb{C}_{\mathrm{z}}^{\mathrm{j,k}}$, we have $\rho^{i} = \rho^{j}$ on $\mathbb{C}_{\mathrm{z}}^{\mathrm{i,j}}$, $\rho^{i} = \rho^{k}$ on $\mathbb{C}_{\mathrm{z}}^{\mathrm{i,k}}$, and $\rho^{j} = \rho^{k}$ on $\mathbb{C}_{\mathrm{z}}^{\mathrm{j,k}}$. If three curves intersect at one point, the point must measure $\rho^{i} = \rho^{j} = \rho^{k}$. That is, the tri-intersection points form a unique path that observe identical pseudoranges to the 3 satellites. The argument can be readily expanded to *m* (*m*>2) pairs of z-curves which intersect.

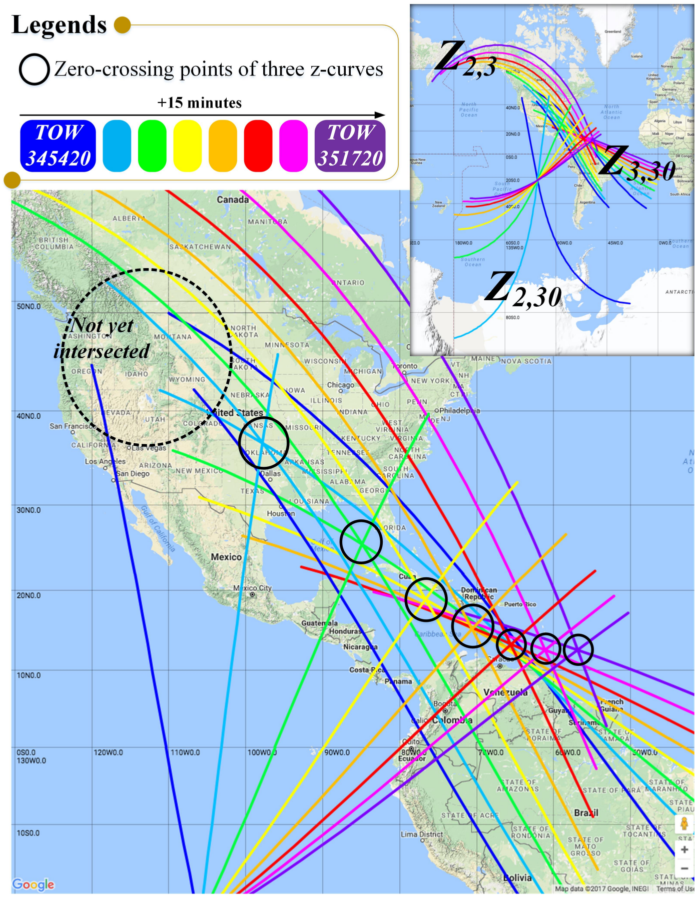


Figure III-8 Intersections of three moving z-curves, Day 110, 2017

III.7 Conclusion

This chapter presents the geometric model and mathematic solution systems for z-surface, z-curve, as well as their direct relatives *k*-surface and *k*-curve. Being a form of TDOA measurement in the 3D space, we present a divide and conquer technique to solve the hyperbolic shapes by (infinite) slicing of the 3D hyperboloid into its 2D slices. A parameterization technique and its two constraint solvers were also presented. Experimental results show that our model match the field data collected from the CORS network. This work is based on a pair of satellites in their fixed positions. Dynamic migration of z-curves and z-surfaces and other dynamic properties need to be developed.

CHAPTER IV

SATELLITE TSV AND COVERAGE FOOTPRINT

IV.1 Introduction

The analysis of a satellite's radiated region is important for design and application of many classes of satellites, e.g. communications, weather, navigation [67], Earth resources, and surveillance satellites. The radiated space of a satellite is known as the *Service Volume*, describing the 3D space reached by the broadcast signal. This radiated region on Earth's surface which receives signals from a given satellite is known as the *coverage footprint*. Geometrically, the service volume is a downward-opening cone-like shape which is a function of the satellite's position and antenna's beam width. An analytical model of service volume is helpful for the design of constellations and satellite antenna patterns.

A formal and validated research of the service volume model is missing in literature. Existing models for coverage footprints have several disadvantages. First, they are point-based numerical solutions [44][47][48]. This class of methods describes the satellite-receiver pair geometry; they discretize the Earth's surface into a set of points and examine the satellite-receiver geometric relationship. The results are less accurate than an analytic solution, and the accuracy depends on the discretization resolution. Second, they bear a high computational complexity - proportional to the number of pixels. If the resolution is less than 1 km^2, there will be millions of ground points for a LEO (Low Earth Orbit) satellite and billions for a MEO (Medium Earth Orbit) satellite. Third, existing methods usually assume the Earth as a sphere for simplicity [42-52]. An ellipsoid is a better approximation of the Earth, with a sphere assumption causes 100-200 kilometers of errors for the circumference of the equator and a meridian circle.

Contributions of this work are twofold. First, I propose an algebraic Cartesian model for the service volume. Adopting an ellipsoidal Earth approximation, this work models the service volume as a cone-based system of quadratic inequalities. With these Cartesian models one can quickly determine if a point is within the service volume, and use them as constraints in optimization problems such as constellation design, antenna parameter selection, etc. Second, this dissertation proposes a parameterization scheme to obtain the boundary of the ground region covered by the service volume. The proposed algorithms derive the boundaries as a list of latitude-longitude points. The resolution of latitudes controls the resolution of boundary points, which reduces the previous 2-dimension computation complexity to 1-dimension. Finally, the proposed theories are validated using GPS satellite ephemeris and real measurements from 946 high-rate observation stations around the globe.

The remainder of the chapter is organized as follows. In Sections IV.2 and IV.3, I derive the Cartesian model for the service volume. Section IV.4 discusses the parameterization scheme for the boundaries of the ground intersections of the service volume. Lastly, in section IV.5, experimental results are presented.

IV.2 Implicit Cartesian Cone Model

I model the TSV of a satellite as a span of a (half) cone shape with its open facing the Earth, as the shaded space illustrated in Figure IV-1 left. I first discuss the half cone model in Lemma IV.1. Then in Theorem IV.1 and Corollary IV.1, we discuss the visibility constraint – how to exclude the shaded area on the dark side of the Earth. The final Cartesian TSV model is provided in Theorem IV.2.

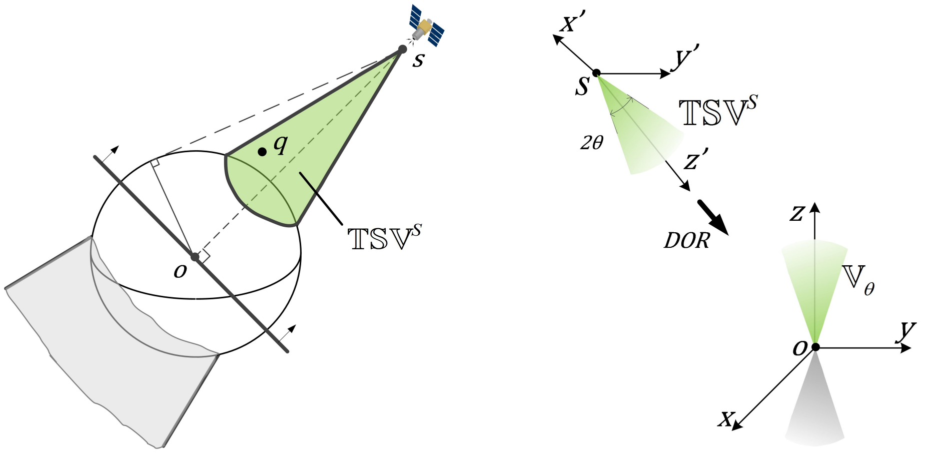


Figure IV-1 Illustrations of the TSV modeling process.

- **Left**: an example TSV shape
- **Right**: transformation process of the TSV cone model

Lemma IV.1: The radiation cone of a satellite S with apex $\mathbf{x}^S = (x^S, y^S, z^S)^T$ and opening angle *2θ*, has this implicit Cartesian expression:

$$\begin{cases} \mathbf{x}^T M_V \mathbf{x} + N_V \mathbf{x} + L_V \leq 0, \\ C_V x + D_V y + E_V z + F_V \geq 0, \end{cases}$$

with coefficients as follows:

$$M_V = \mathcal{R}_x(-\alpha)\, \mathcal{R}_y(\beta) diag(1,1,-tan^2\theta) \mathcal{R}_y(\beta)^T \mathcal{R}_x(-\alpha)^T,$$

$$N_V = -[M_V \mathcal{T} + M_V^T \mathcal{T}]^T,\ L_V = \mathcal{T}^T M_V \mathcal{T},$$

$$C_V = x^P - x^S,\ D_V = y^P - y^S,\ E_V = z^P - z^S,$$

$$F_V = \mathbf{x}^{S^T}(\mathbf{x}^S - \mathbf{x}^P),$$

where $\mathcal{T}$, $\mathcal{R}_x(-\alpha)$, and $\mathcal{R}_y(\beta)$ are a translation vector, and two rotation matrices about the x and y-axes to be given in (IV.3)-(IV.6).

Proof: The modeling process of 𝕋𝕊𝕍ˢ starts with introducing a canonical cone $\mathbb{V}_\theta$ at the ECEF origin, as shown in Figure IV-1 right, aligned with the z-axis, and with opening angle 2θ. Let $\mathbf{x}_0 \in \mathbb{V}_\theta$ denote an interior point of $\mathbb{V}_\theta$. All such points $\mathbf{x}_0$ will be transformed to be interior points $\mathbf{x}$ ∈𝕋𝕊𝕍ˢ, with the following transition:

$$\mathbf{x} = \mathcal{R}\mathbf{x}_0 + \mathcal{T}, \tag{IV.1}$$

where $\mathcal{R}$ and $\mathcal{T}$ are a 3×3 rotation matrix and a 3×1 translation vector to be discussed later.

The source cone - $\mathbb{V}_\theta$ in use here from be truncated from a canonical double cone (including the gray downward-looking cone in Figure IV-1 right) by the positive z-axis component [53]:

$$\mathbb{V}_\theta \colon \begin{cases} \mathbf{x}_0^T M_\theta^{cone} \mathbf{x_0} \leq 0, \\ z_0 > 0, \end{cases} \tag{IV.2}$$

where $M_\theta^{cone} = diag(1,1,-tan^2\theta)$ is a coefficient characterizing a canonical double cone, and the "$z_0 > 0$" term is added to exclude the negative cone (in gray). The target cone - 𝕋𝕊𝕍ˢ has $\mathbf{x}^S$ as apex, pointing towards the nadir p in the direction of radiation, denoted as $\mathbf{d}$:

$$\mathbf{d} = (\frac{x^P - x^S}{||\mathbf{x}^P - \mathbf{x}^S||}, \frac{y^P - y^S}{||\mathbf{x}^P - \mathbf{x}^S||}, \frac{z^P - z^S}{||\mathbf{x}^P - \mathbf{x}^S||})^T,$$

where $||\mathbf{x}^P - \mathbf{x}^S||$ is the distance between p and S, serving as the normalizing factor.

Transformation from $\mathbf{x}_0 \in \mathbb{V}_\theta$ to $\mathbf{x} \in$ 𝕋𝕊𝕍ˢ is accomplished in (IV.1) by first rotating the positive z direction to align the $\mathbf{d}$ (Eqns. (IV.3)-(IV.5)), followed by translation $\mathcal{T}$ of the shape from $(0,0,0)^T$ to $\mathbf{x}^S$ (Eqn. (IV.6)).

The rotation process is illustrated in Figure 4, where the target vector $\mathbf{d} = (d_x, d_y, d_z)^T$ is placed on the origin, and it is to be rotated to align with the z-axis along intermediate vectors

$(d_x, 0, \sqrt{d_y^2 + d_z^2})^T$ on the xz-plane with angle α, and $(0,0, \left||\mathbf{d}|\right|^2 = 1)^T$ on the z-axis with angle β. Note that in this process, the length of all intermediate vectors is preserved to be identical to that of **d**, i.e. a unit length. The rotation from the z-axis $(0,0,1)^T$ to **d** includes a counter-clockwise rotation of β about the y-axis, and a clockwise rotation of α about the x-axis. Rotations of arbitrary angles ω_x and ω_y about x- and y-axes on a 3D point can be done through left multiplying it with the following matrices, respectively:

$$\mathcal{R}_x(\omega_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\omega_x & -\sin\omega_x \\ 0 & \sin\omega_x & \cos\omega_x \end{pmatrix}, \quad \mathcal{R}_y(\omega_y) = \begin{pmatrix} \cos\omega_y & 0 & \sin\omega_y \\ 0 & 1 & 0 \\ -\sin\omega_y & 0 & \cos\omega_y \end{pmatrix}. \quad \text{(IV.3)}$$

Specifically, we need $\omega_x = -\alpha$ and $\omega_y = \beta$ because $\mathcal{R}_x$ and $\mathcal{R}_y$ are counterclockwise from the perspective of a positive axis toward the origin [54]. As such, the rotation matrix in (IV.1) is $\mathcal{R} = \mathcal{R}_x(-\alpha)\ \mathcal{R}_y(\beta)$, indicating that the point $\mathbf{x}_0$ shall be rotated about y-axis before rotation about x-axis.

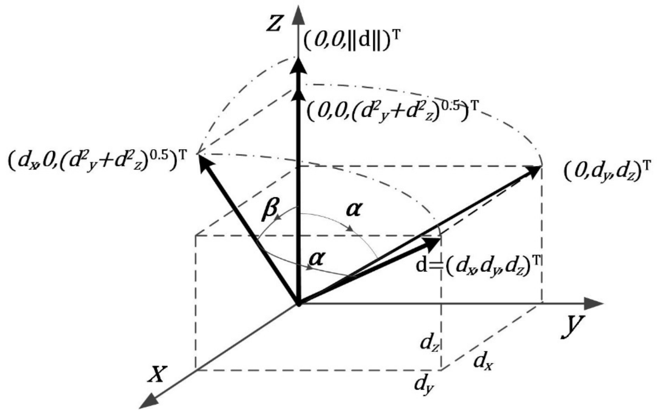


Figure IV-2 Angles from z-axis to d in the ECEF coordinate frame.

The angle β can be readily calculated by considering the sides of the right triangle at the origin, $(d_x, 0, \sqrt{d_y^2 + d_z^2})^T$, and $(0,0,\sqrt{d_y^2 + d_z^2})^T$:

$$\cos\beta = \frac{\sqrt{d_y^2+d_z^2}}{||\mathbf{d}||} = \sqrt{d_y^2 + d_z^2}, \quad \sin\beta = \frac{d_x}{||\mathbf{d}||} = d_x. \qquad \text{(IV.4)}$$

The angle α is calculated on the yz-plane, because α is equal to the angle between the z-axis and $\mathbf{d}$'s projection on the yz-plane $(0, d_y, d_z)^T$:

$$\cos\alpha = \frac{d_z}{\sqrt{d_y^2+d_z^2}}, \sin\alpha = \frac{d_y}{\sqrt{d_y^2+d_z^2}}. \qquad \text{(IV.5)}$$

In (IV.4), there are odd cases when the denominator $\sqrt{d_y^2 + d_z^2}$ is zero: $d_y = d_z = 0$. In this case when the $\mathbf{d}$ is already on the x-axis, there is no need to rotate about x-axis, and a single rotation of β about y-axis is sufficient.

The translation process is equivalent to translating the origin to the position of the satellite S. The translation vector $\mathcal{T}$ in (IV.1) can be constructed by taking the difference of these two points, or simply [54]:

$$\mathcal{T}(\mathbf{x}^S) = (x^S, y^S, z^S)^T. \qquad \text{(IV.6)}$$

The Cartesian representation of $\mathbb{TSV}^{\mathrm{S}}$ cone is obtained by rewriting (IV.1) into its inverse function, $\mathbf{x}_0 = \mathcal{R}^{-1}(\mathbf{x} - \mathcal{T})$, and substituting it into the quadratic inequality in (IV.2). Details of the derivation process is left to the readers, who might need to utilize the property that the rotation matrices $\mathcal{R}_x$ and $\mathcal{R}_y$ are orthogonal, i.e.${\mathcal{R}_x}^T = {\mathcal{R}_x}^{-1}$ and ${\mathcal{R}_y}^T = {\mathcal{R}_y}^{-1}$. The resultant expression is present as follows:

$$\mathbf{x}^T M_V \mathbf{x} + N_V \mathbf{x} + L_V \leq 0, \qquad \text{(IV.7)}$$

where $M_V = \mathcal{R} M_\theta^{cone} \mathcal{R}^T$, $N_V = -[M_V \mathcal{T} + M_V^T \mathcal{T}]^T$, $L_V = \mathcal{T}^T M_V \mathcal{T}$.

The “$z_0 > 0$” in (IV.2) is transformed to a plane intersecting S and with $\mathbf{d}$ as its normal vector. Based on the general point-normal form of a 3D plane [55], one can readily represent the transformed plane “$z_0 > 0$” as:

$$C_V x + D_V y + E_V z + F_V \geq 0, \tag{IV.8}$$

where $C_V = x^P - x^S$, $D_V = y^P - y^S$, $D_V = z^P - z^S$, $F_V = \mathbf{x}^{S^T}\mathbf{x}^S - \mathbf{x}^{S^T}\mathbf{x}^P$. ■

IV.3 Visibility Constraints

Points which are not visible to satellites are excluded via the visibility constraint. For instance in Figure IV-3 left, the dark side of the Earth should not be modeled within the TSV. However, the modeling process is not as simple as using a plane to exclude the dark side of the Earth. Experiments have shown inclusion of invalid space commonly occurs in MEO (medium earth orbit), and GEO (geostationary earth orbit) satellites, as shown in Figure IV-3 left. The shaded areas should be excluded from the TSV model, when the MEO or higher satellites are designed to cover the entire visible Earth surface. Therefore, a more rigorous visibility constraint is required.

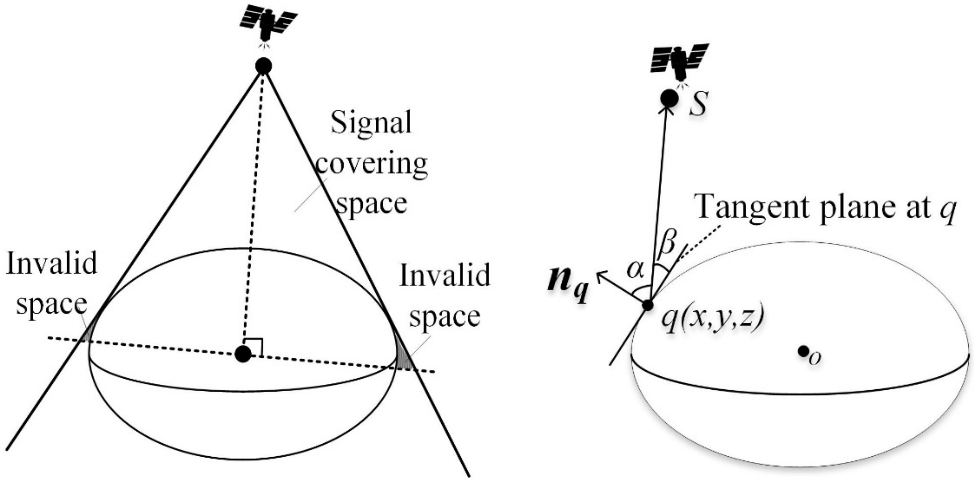


Figure IV-3 Modeling process of visibility constraints

- **Left**: blind spots of the naïve visibility constraint
- **Right**: geometry of horizontal coordinates

Inspired by the word "visibility", our modeling process focuses on the satellite's elevation – the satellite should be above the horizon of the ground receiver. Let q denote an arbitrary point on the Earth surface. We are interested in the set of q's which view the satellite S at a non-negative elevation.

As illustrated in Figure IV-3 right, β denotes the elevation angle of satellite S with respect to its local horizon (i.e. the tangent plane to the Earth), and α the complement angle of β. The angle α can be computed by the normal vector $\mathbf{n}_q$ of the tangent plane and the vector $\boldsymbol{s}$ from q to S, denoted by $\boldsymbol{qs}$: $\cos\alpha = \frac{\boldsymbol{qs}\cdot\mathbf{n}_q}{|\boldsymbol{qs}||\mathbf{n}_q|}$. Vector $\mathbf{n}_q$ at point q, is obtained by simply taking partial derivatives of the Earth equation (II.15) with respect to x, y, z:

$$\mathbf{n}_q = \left(\frac{2x}{a_E^2}, \frac{2y}{a_E^2}, \frac{2z}{b_E^2}\right)^T. \tag{IV.9}$$

We want all possible q's whose elevation is non-negative, or specifically, $\beta \geq \beta_0$, where β_0 is a minimum elevation angle mask, β_0 is a constant, and β is a function of q. We show the derivation process in detail (symbol "→" indicates "derives to"):

$\beta \geq \beta_0 \ \rightarrow\ \sin\beta \geq \sin\beta_0 \ \ \rightarrow\ \sin(\frac{\pi}{2} - \alpha) \geq \sin\beta_0 \ \rightarrow\ \ \cos\alpha \geq \sin\beta_0 \ \ \rightarrow$

$$\boldsymbol{qs} \cdot \mathbf{n}_q \geq \sin\beta_0 \,|\boldsymbol{qs}||\mathbf{n}_q|. \qquad \text{(IV.10)}$$

Based on the inequality (IV.10), we can derive the following two theorems about satellite visibility constraints.

Theorem IV.1: Given an elevation angle β_0 ($\beta_0 \in [0, \frac{\pi}{2}]$), all ground points $\mathbf{x}$ who view satellite S $\mathbf{x}^S = (x^S, y^S, z^S)^T$at an elevation angle no smaller than β_0 (i.e. $\beta \geq \beta_0$) can be expressed as:

$$\mathbf{x}^T M_G \mathbf{x} + N_G \mathbf{x} + L_G \geq 0,$$

where $M_G = \begin{pmatrix} \frac{x^{S^2}}{a_E^4} - sin^2\beta_0 & \frac{x^S y^S}{a_E^4} & \frac{x^S z^S}{a_E^2 b_E^2} \\ 0 & \frac{y^{S^2}}{a_E^4} - sin^2\beta_0 & \frac{y^S z^S}{a_E^2 b_E^2} \\ 0 & 0 & \frac{z^{S^2}}{b_E^4} - sin^2\beta_0 \end{pmatrix}$,

$$N_G = (2x^S sin^2\beta_0 - \frac{x^S}{a_E^2}, 2y^S sin^2\beta_0 - \frac{y^S}{a_E^2}, 2z^S sin^2\beta_0 - \frac{z^S}{b_E^2}),$$

$$L_G = 1 - sin^2\beta_0(x^{S^2} + y^{S^2} + z^{S^2}).$$

Proof: The proof process is to expand inequality (IV.10), where $\boldsymbol{qs} = (x^S - x, y^S - y, z^S - z)^T$ and n_q is given in (IV.9).

In the LHS (Left-Hand Side), we have

$$\boldsymbol{qs} \cdot \mathbf{n}_q = \frac{2x}{a_E^2}(x^S - x) + \frac{2y}{a_E^2}(y^S - y) + \frac{2z}{b_E^2}(z^S - z)$$

$$\boldsymbol{qs}\cdot\mathbf{n}_q = \frac{2x}{a_E^2}x^S + \frac{2y}{a_E^2}y^S + \frac{2z}{b_E^2}z^S - 2\left(\frac{x^2}{a_E^2} + \frac{y^2}{a_E^2} + \frac{z^2}{b_E^2}\right)$$

$$\boldsymbol{qs}\cdot\mathbf{n}_q = \frac{2x^S}{a_E^2}x + \frac{2y^S}{a_E^2}y + \frac{2z^S}{b_E^2}z - 2\ .$$

In the RHS (Right-Hand Side), we have

$$|\boldsymbol{qs}| = \sqrt{x^2 + y^2 + z^2 - 2x^Sx - 2y^Sy - 2z^Sz + {x^S}^2 + {y^S}^2 + {z^S}^2} \text{ and}$$

$$\left|\mathbf{n}_q\right| = 2\sqrt{\frac{x^2}{a_E^4} + \frac{y^2}{a_E^4} + \frac{z^2}{b_E^4}} = 2.$$

Since $0 \le \beta_0 \le \frac{\pi}{2}$, squaring two sides of this inequality (IV.10) should still be satisfied, such that the square root in $|\boldsymbol{qs}|$ can be removed:

$$(\boldsymbol{qs}\cdot\mathbf{n}_q)^2 \ge (2\sin\beta_0\,|\boldsymbol{qs}|)^2,$$

which can be derived to matrix form shown in Theorem IV.1, by substituting in equations above.

■

Theorem IV.1 can be used to plot and analyze the relationship of satellite position and ground regions with different elevation viewing angles. For example, GPS/GNSS receivers are encouraged to ignore satellites that are flying at 5° or lower elevations for better signal quality. In our discussion, we are more interested in "visible or not", so the following corollary provides the visibility constraint by setting $\beta_0 = 0$.

Corollary IV.1: All ground points **x** who view satellite S $\boldsymbol{x}^S = (x^S, y^S, z^S)^T$ above or on their horizon (i.e. $\beta \ge 0$) can be expressed as:

$$\frac{x^S}{a_E^2}x + \frac{y^S}{a_E^2}y + \frac{z^S}{b_E^2}z \ge 1.$$

Proof: Based on Theorem IV.1, we set $\beta_0 = 0$ and obtain the above planar inequality. The derivation process is omitted for space. ■

In summary, I can finally express the Cartesian model for a TSV in the following Theorem.

Theorem IV.2: The TSV of a satellite S, $\mathbb{TSV}^{S}$, about the reference ellipsoid defined in WGS84 has the following implicit Cartesian expression:

$$\begin{cases} \mathbf{x}^T M_V \mathbf{x} + N_V \mathbf{x} + L_V \le 0, \\ C_V x + D_V y + E_V z + F_V \ge 0, \\ \mathbf{x}^T M_E \mathbf{x} + N_E \mathbf{x} + L_E \ge 0, \\ \frac{x^S}{a_E^2} x + \frac{y^S}{a_E^2} y + \frac{z^S}{b_E^2} z - 1 \ge 0, \end{cases} \tag{IV.11}$$

where M_V, N_V, L_V are defined in Lemma IV.1, C_V, D_V, E_V, F_V in Lemma IV.1 as well, M_E, N_E, L_E from (1).

Proof: As I have discussed in Lemma IV.1, the 1st and 2nd inequalities of (IV.11) restrict the points to be inside of the half cone. The 3rd inequality restricts points to be on or outside of the Earth's surface. The 4th inequality ensures points visible to satellite, as given in Corollary IV.1. ■

IV.4 Parameterization for Footprint Boundary

This section constructs the parametric function $P(\lambda, \varphi) = 0$, with whom only parametric coordinates (λ, φ) on the boundary of TSV footprint are satisfied. This section also discusses the solving process for all qualified λ's, given a value of the tracing parameter φ. The reasonable search range of the tracing parameter is provided as well.

Generally, the footprint boundary is modeled as an intersection of the TSV cone and the Earth ellipsoid. As a Quadratic Surfaces Intersection Curve (QSIC), it should be a geodesic surface ellipse, as illustrated in Figure IV-4 left. However, corner cases may break this perfect nature, as shown in Figure IV-4 right. When the TSV cone does not intersect with the Earth completely, the resulting footprint boundary will not be a perfect ellipse. Instead, a desired footprint boundary is plotted in Figure IV-5, where it is composed of two parts: cone's intersection with Earth, as well as the visibility boundary. In the following paragraphs I discuss how to trace each.

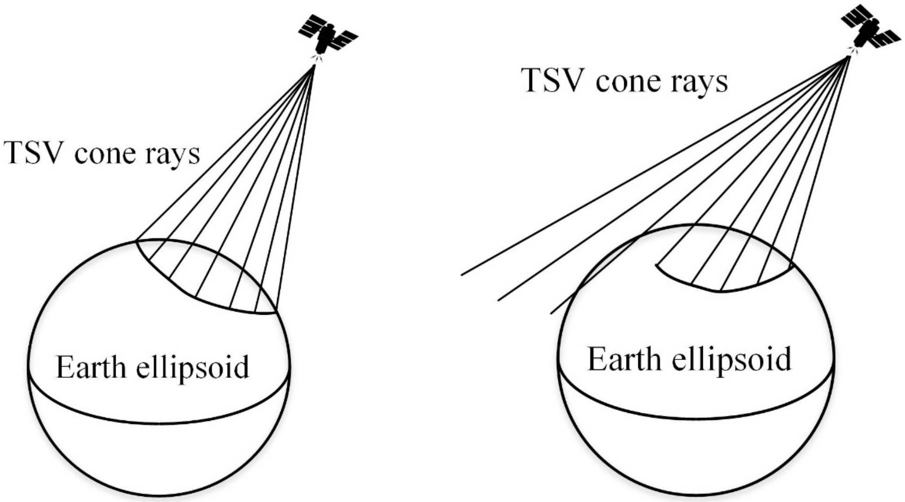


Figure IV-4 Two intersections between TSV cone and Earth ellipsoid

- **Left**: cone completely intersects with ellipsoid
- **Right**: cone incompletely intersects with ellipsoid

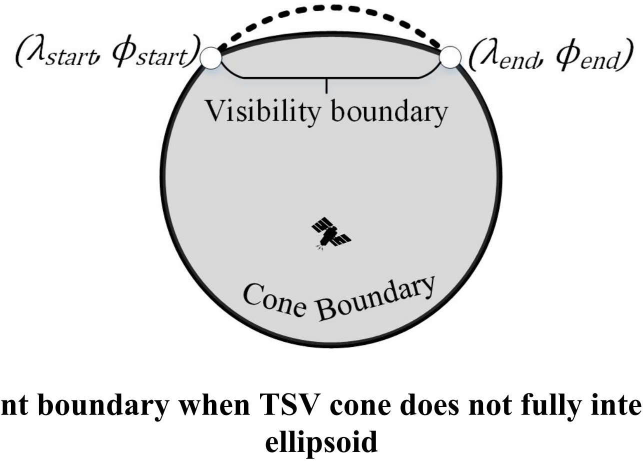

$(\lambda_{start}, \phi_{start})$
$(\lambda_{end}, \phi_{end})$
Visibility boundary
Cone Boundary

Figure IV-5 Footprint boundary when TSV cone does not fully intersect with the Earth ellipsoid

The *cone boundary* assumes the TSV cone fully intersected with the Earth ellipsoid, as shown in Figure IV-4 left. In this case, the footprint boundary would appear like the circle in Fig. 7, including the dash part. Let $P_{cone}(\lambda, \varphi) = 0$ denote the parametric function for this case. Both the ellipsoid and cone have parametric forms in terms of parameters other than x, y, z. Here we adopt the parametric cone, because a cone is a *ruled surface* (i.e. a surface that be represented by a cluster of rotating lines) that will lead to a simpler P_{cone} function.

Specifically, let $\mathbf{x} = \mathbf{x}^S + t \cdot \mathbf{v}$ ($t \in \mathbb{R}, t > 0$) denote all rays on the $\mathbb{TSV}^{\mathrm{S}}$ cone's

boundary originating from the satellite S, towards the Earth, in a unit direction $\mathbf{v}$. Since every $\mathbf{v}$ is of unit-length, and has a θ-angle with respect to the satellite's Direction of Radiation (DOR) $\mathbf{d}$ (defined in the proof of Lemma IV.1), we can calculate all $\mathbf{v}$'s as finely as required through selecting all $v_z \in [-1,1]$ and solving

$$\begin{cases} v_x^2 + v_y^2 + v_z^2 = 1, \\ \mathbf{v} \cdot \mathbf{d} = \cos\theta\,. \end{cases}$$

Then $\mathbf{x} = \mathbf{x}^S + t \cdot \mathbf{v}$ is substituted into the Earth's Cartesian expression (II.15), leading to a quadratic single-variable (t) equation. If no real solution exists, we skip this ray because it implies this ray does not intersect with the Earth. If one or two real positive solutions exist, the smallest value of t is chosen, because it is closer to the satellite. With known t and $\mathbf{v}$, the boundary points on the coverage zone can be found out in both Cartesian and (λ, φ) coordinates.

The *visibility boundary* is the intersection of the visibility constraint (defined in Corollary 1) and the Earth surface. It is the intersection of a plane and an ellipsoid. We can plug the parametric Earth into the visibility plane ($\frac{x^S}{a_E^2}x + \frac{y^S}{a_E^2}y + \frac{z^S}{b_E^2}z = 1$).

$$C_\varphi \cos(\lambda) + D_\varphi \sin(\lambda) + E_\varphi = 0, \qquad \text{(IV.12)}$$

where $C_\varphi, D_\varphi, E_\varphi$ are functions of the tracing parameter φ:

$$C_\varphi = -\frac{x^S}{a_E}\cos(\varphi)\,, D_\varphi = -\frac{y^S}{a_E}\cos(\varphi)\,, E_\varphi = 1 - \frac{z^S}{b_E}\sin(\varphi).$$

The solving process of (IV.12) is explained in Appendix 2.

IV.5 Experimental Validation & Analysis

In this section, I carry out experiments to validate the correctness of the Cartesian models of the TSV and the parameterization schemes for the footprint boundary. I then use the proposed models to analyze the service coverage of some popular constellations.

Validation

I use GPS to validate the proposed models, because research resources for GPS are very rich and accessible publicly online. There are over 946 high-rate GPS/GNSS reference stations by the International GNSS Service (IGS) network and the Continuously Operating Reference Station (CORS) networks over the globe [23][27]. Stations constantly report raw measurements at every second for every GPS satellite. These reports are collected in public FTP servers and used as ground truth to indicate whether stations observe GPS signal.

The validation experiment is implemented in *C++* and *Javascript*. The process is as follows. First, I compute the ECEF position ($\mathbf{x}^S$) of a GPS satellite using the broadcasted ephemeris model from NASA. It results in up 1 meter of satellite position error. Second, I construct the Cartesian TSV model with GPS beam width angle ($2\theta = 27.6°$), using Theorem IV.2. Third, I trace its footprint boundary using the proposed parameterization scheme, and plot the boundary on *Google Maps*. Fourth, on the same map, I show all high-rate reference stations and mark those who report measurements within 5 seconds from the GPS satellite.

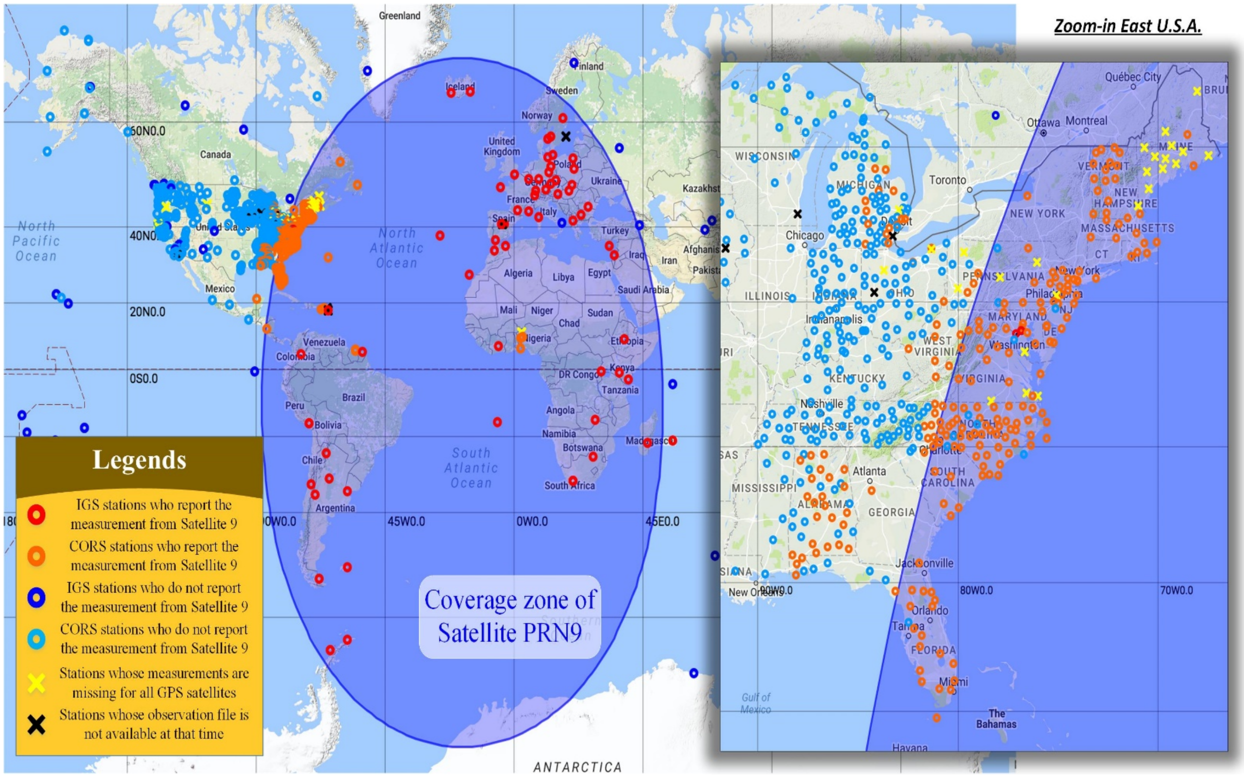

Zoom-in East U.S.A.
Coverage zone of
Satellite PRN9
Legends
IGS stations who report the measurement from Satellite 9
CORS stations who report the measurement from Satellite 9
IGS stations who do not report the measurement from Satellite 9
CORS stations who do not report the measurement from Satellite 9
Stations whose measurements are missing for all GPS satellites
Stations whose observation file is not available at that time
60N0.0
40N0.0
20N0.0
0S0.0
0W0.0
45W0.0
0W0.0
45E0.0
90W0.0
80W0.0
70W0.0
North Pacific Ocean
North Atlantic Ocean
South Atlantic Ocean
ANTARCTICA
Greenland

Figure IV-6 Validation of the proposed TSV model (global view and zoom-in view of East U.S.A.).

Note: The blue shaded oval is the coverage zone of PRN 9, at TOW 352700 second, Apr 20, 2017, produced by the parameterization method.

The experimental result is shown in Figure IV-6. The footprint boundary of GPS satellite PRN (Pseudorandom Number) 9 is generally matched with the boundary between unreported stations (in blue or light blue) and reported stations (in red or orange). To have closer observations, I zoom into the dense area in North America, as shown in Figure IV-6 right. It shows the proposed model and parameterization scheme provide an accurate approximation. Most reference stations report measurement as soon as they see the satellite. Some stations inside the coverage have not done so because they are configured to report satellites with 5° or higher elevation. Interestingly, over half of stations located in the state of Alabama, and some stations in the state of Michigan report measurements even when they have not seen the satellite. This is due to the use of highly sensitive Trimble receivers, capable of receiving satellite signals from negative elevations.

Analysis

This section uses the proposed TSV model and parameterization method to analyze coverage of different constellations. The output from our work is a list of latitude-longitude pairs to denote the coordinates of footprint boundary. To calculate the coverage area on the Earth ellipsoid, we utilize the "*areaint*" function of the Matlab Mapping toolbox [63]. It uses a line integral approach to add up the total polygon areas constraint by the footprint boundary, on a WGS84 reference ellipsoid.

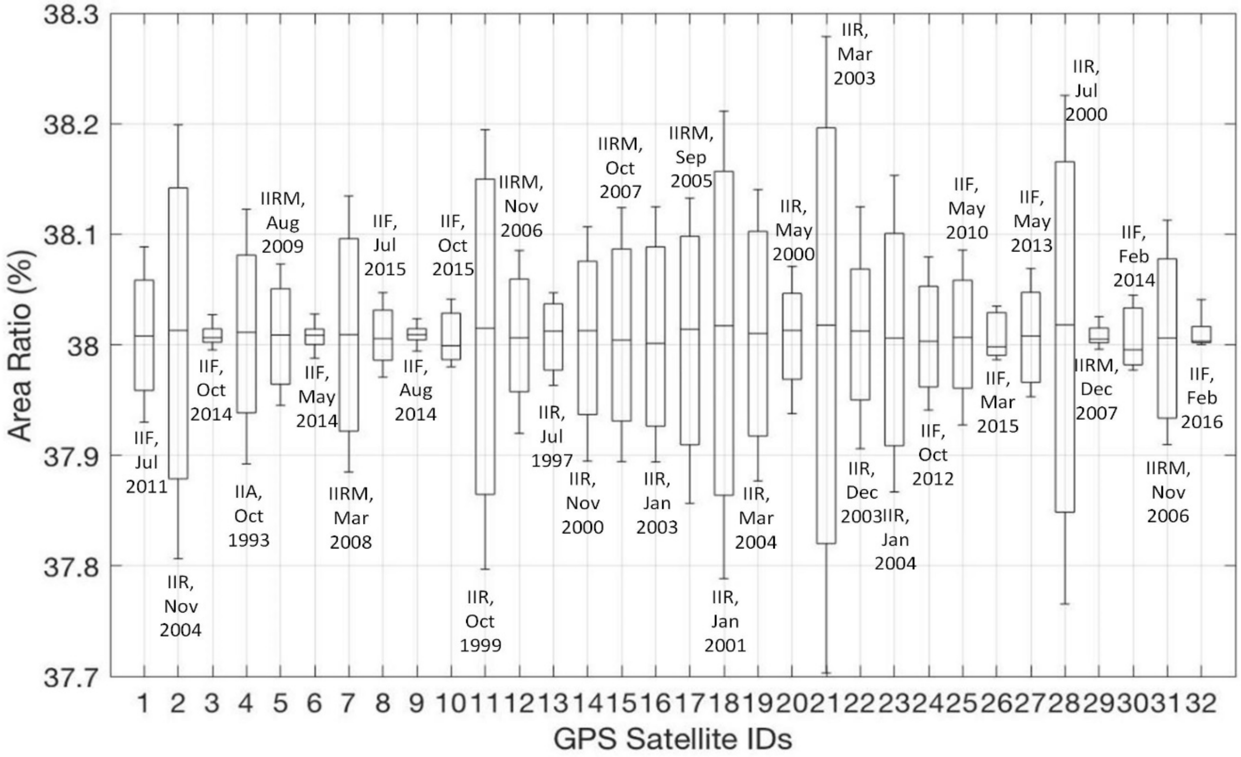


Figure IV-7 Coverage area ratio on the reference ellipsoid of GPS satellites, annotated with satellites' block type and launch date

The ratio of each GPS satellite's footprint area over the entire area of the Earth (510072000 km^2) is shown in Figure IV-7. The x-axis has 32 bins, referring to GPS satellites with PRNs 1 – 32. The y-axis is the area ratio, for the reference of the minimal, 25%-quantile, half, 75%-quantile, and the maximal of values over one GPS ECEF period (about 23 hours 56 minutes). Overall, the coverage ratio of GPS is around 38% with ±0.2% uncertainty. The uncertainty may be caused by the orbital shape: coverage area is the maximum when a satellite is at apogee, or the minimum when it is at perigee. Moreover, newly launched satellites (e.g. PRNs 3, 6, 9, 32) tend to have less uncertain coverage ratios than older satellites (e.g. PRNs 2, 11, 18, 21, 28).

In Figure IV-8, we continue to use GPS satellite positions to examine the assumption of "spherical Earth" versus "ellipsoidal Earth". Specifically, we change the y-axis to the area difference between the footprint on Earth ellipsoid (WGS84) and the footprint on Earth sphere (with 3 different radius). Assuming the Earth radius to be 6371 km, the resulting coverage area has an error of 48,000 km^2 (equivalent to the land area of Slovakia [62]) to footprint on a reference ellipsoid. If we select the radius to be the lengths of semi-major axis a_E or semi-minor axis b_E, the area errors will be around 73,000 km^2 (the land area of Panama) and 140,000 km^2 (the land area of Nepal). Future work includes determination of the optimal radius of the Earth for coverage analysis, which requires a more focused study to consider other factors.

In Table IV-1, I summarize the mean and uncertainty of some constellations' altitude and coverage area, computed by the proposed works. Positions of all satellites in each constellation are calculated using Two-Line-Element orbital models from *Celestrak* [65] and the SGP4 v1.4 Python library [66]. Accuracy is up to 1 kilometer within one day.

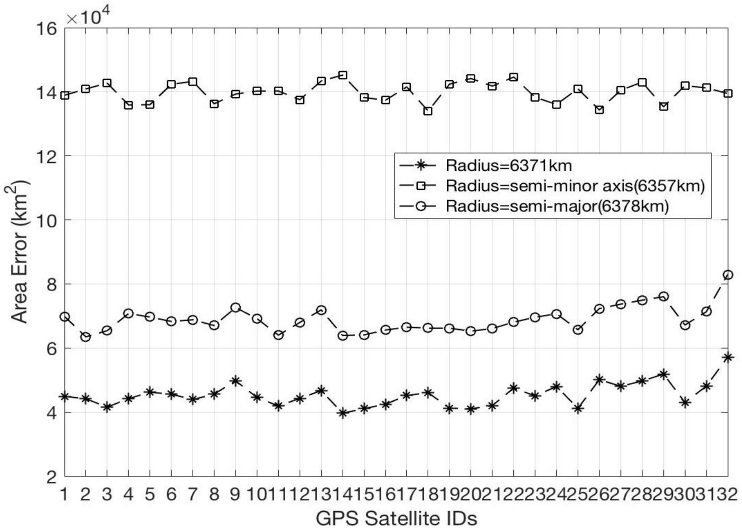


Figure IV-8 Errors of coverage areas with spherical Earth versus ellipsoidal Earth

Table IV-1 Constellation coverage comparisons

Category	Constellation	Operator	Purpose, Number of Operational Satellites	Orbital Period	Off-Nadir Angle (θ)	Mean Altitude (km)	Altitude Uncertainty (km)	Mean Coverage Ratio	Mean Coverage Area (km^2)	Area Uncertainty (km^2)
Low Earth Orbit (LEO)	Orbcomm	Orbcomm (United States)	Communication, 44	~100 minutes [36]	62°[33]	745.73	48.71	5.23%	26,663,351.38	1,565,543.80
	Iridium	Iridium Communications Inc.	Communication, 72	~100 minutes [35]	75°[34]	693.81	167.61	4.87%	24,832,541.76	5,704,427.89
	Iridium-NEXT	Iridium Communications Inc.	Communication and Earth survey, 66	~100 minutes [35]	75°[34]	786.00	22.56	5.47%	27,920,337.50	734,212.92
Medium Earth Orbit (MEO)	GPS	United States Air Force	Navigation, 32	11 hours 58 minutes [29]	13.8°[31]	20,191.41	191.29	38.01%	193,865,417.85	447,872.81
	Galileo	European GNSS Agency (GSA)	Navigation, 30	14 hours 22 minutes [29]	13.8°	23,116.08	923.46	39.19%	199,874,833.27	2,002,361.44
	GLONASS	Russian Federal Space Agency	Navigation, 25	11 hours 15 minutes [29]	13.8°	19,139.18	24.20	37.52%	191,367,305.14	136,304.75
	Beidou	China National Space Administration (CNSA)	Navigation, 18	12 hours 53 minutes [30]	13.8°	21,543.89	40.58	38.59%	196,830,718.91	109,153.00
Geostationary / Geosynchronous Earth Orbit (GEO)	SBAS	U.S.A., E.U., Japan, India, Russia	Navigation Augmentation, 16		9°[32]	35,801.09	775.28	42.43%	216,439,687.99	708,942.89
	Gorizont	Russia	Communication, 35		9°	36,069.56	1,835.34	42.47%	216,623,743.10	1,786,604.03

IV.6 Conclusion

This chapter presents a geometric Cartesian model for the TSV from a satellite to its nadir with any specified off-nadir angle (Theorem IV.2). This chapter also provides a parameterization method to obtain the discrete points on the satellite’s footprint boundary. This footprint boundary can be the horizon edge (Corollary IV.1) or any β-degree elevation edge (Theorem IV.1) for ground points to the satellite. Experiments with real GPS ephemeris show that our model and parameterization method match the field measurements from CORS/IGS networks. I then use the proposed theories to analyze coverage of GPS along with other constellations.

CHAPTER V

DYNAMIC MODELS OF TSV, Z-SURFACE & Z-CURVE

V.1 Problem Statement

The z-surfaces formed by pairs of satellite signals migrate constantly in accordance to the satellite motions. Z-surface offers a new tool for periodic integrity and precision checking of signals based on simple observations of signal pairs. Z-surface finds applications in detection of aberrant GNSS signals, e.g., satellite malfunctions, spoofing, multipath. And when the system is absent of such anomaly, their occurrences can be used as a high confidence timing reference for wide-area systems like power transmission grids.

In my prior works, I have demonstrated detection of a z-surface, and later derived the analytical form for z-surface and their parameterization scheme at a time instant. Knowing that the z-surfaces constantly migrate with satellite motions, as the example illustrated in Figure V-1, it is of great interest to model the dynamics of z-surfaces as a function of the satellite speeds. Other basic information include the z-surface/curve motion direction, the shape and pose (flattening, extension, and inclination) of z-surface and z-curve as a function of time. This way, one can generate short-term z-surface forecast for better protection of the system integrity.

In this chapter, I aim to solve the dynamics of the z-surface and z-curve as a function of time for a pair of satellites. Specifically, given the z-surface and z-curve at a time instant $t,$ I aim to solve its shape and position changes at $t + \Delta t$, where Δt is a discrete time interval. A continuous analytic function for shape migration is not considered here, because the orbital model – ephemeris, is a discrete model for the unsmooth orbit. A naïve approach to solve this problem could be based on plugging the new time stamps in static models in Chapter III, but this approach provides little insight on the behavior of the system dynamics. Instead, in this chapter, I propose a system dynamics modeling approach to derive the transformation functions $\mathcal{P}_V$ and $\mathcal{P}_Z$, where the former represents the motion effects of the TSV of a satellite, and the latter motion effects of the *z*-surface. I then pay special attention to the migration of the parameterized version of z-curve, for which, I propose a novel numerical algorithm to move all z-curve points from t to $t + \Delta t$.

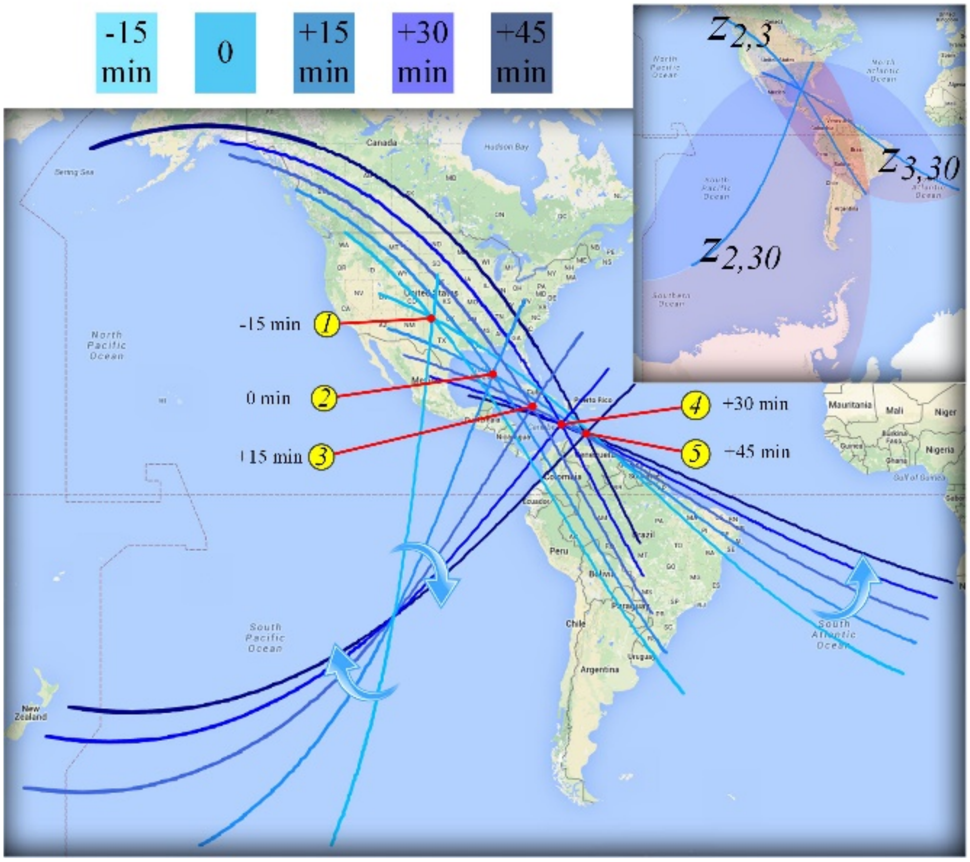


Figure V-1 Moving z-curves of GPS satellites (PRNs 2, 3, 30) with 15-minute interval.

Given the positions of ($\mathcal{A}$, $\mathcal{B}$), one can directly write down the Cartesian representations of their TSVs (denoted as $\mathbb{TSV}^{A,B}$), z(k)-surface ($\mathbb{S}^{A,B}_{z|k}$), and z(k)-curve ($\mathbb{C}^{A,B}_{z|k}$), as well as the parameterized form of z(k)-curve ($\mathbf{C}_{z|k}^{A,B}$) in terms of tuples of latitude-longitude (or equivalently the *Earth-Centered Earth-Fixed* (ECEF) coordinates). $\mathbf{C}_{z}^{A,B}$ is a discretized array derived from $\mathbb{C}_{z}^{A,B}$, and it can be expressed as $\mathbf{C}_{z}^{A,B} = \{\mathbf{r}_i\}, \forall i = \{1,2,\dots,n\}$, where $\mathbf{r} = (x,y,z,1)^T$ is homogeneous coordinate with first three dimensions as the ECEF coordinate and the last as the point's scale. The homogeneous coordinate system is commonly used to simplify transformations.

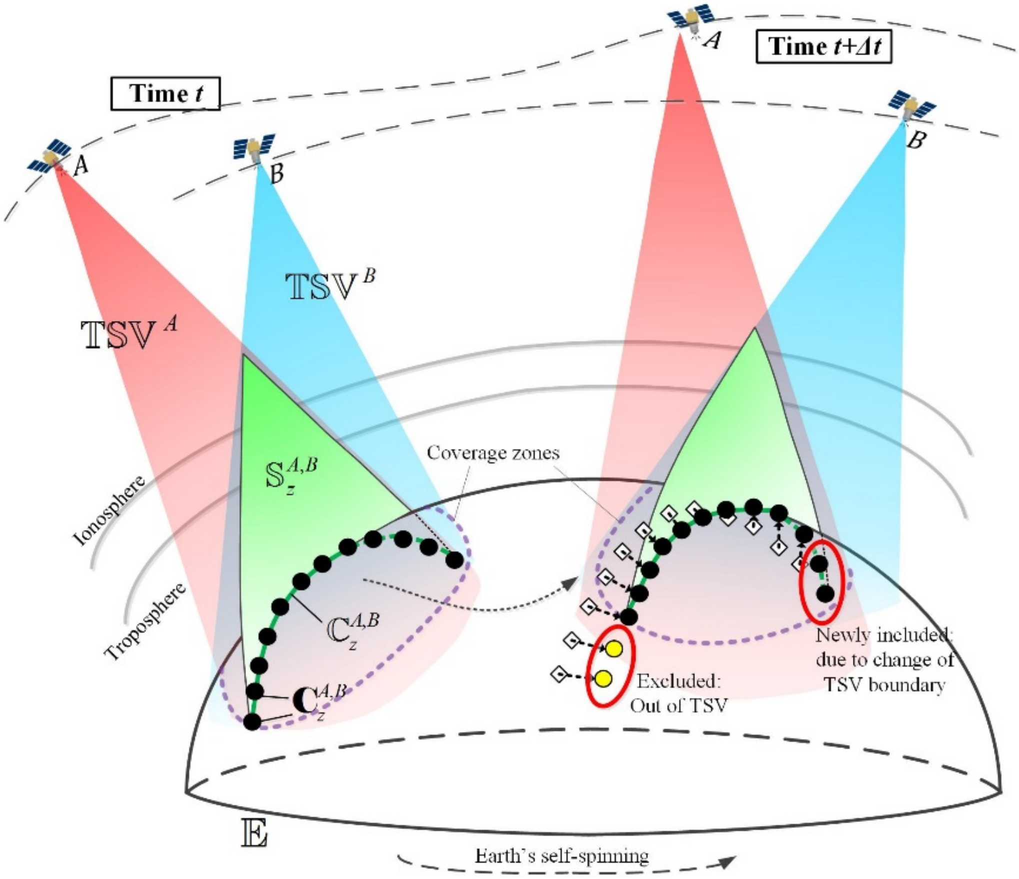


Figure V-2 Illustration of TSV and z-curve migrations

Changes of z-curves are the direct results of satellite and Earth motions. As shown in Figure V-2, during Δt the Earth is self-spinning, ($\mathcal{A}$, $\mathcal{B}$) orbit in different velocities and trajectories, and their signals propagate through the atmosphere from new angles. As a result, the coverage zone of ($\mathcal{A}$, $\mathcal{B}$) and the z-surface morph in terms of shape and displacement. I will show that, through analysis of those motion dynamics, these motion related effects can be summarized into 4×4 matrices $\mathcal{P}_V$ and $\mathcal{P}_z$ plus filtering operations of boundary:

$$\mathbf{r}_V(t + \Delta t) = \mathcal{P}_V \mathbf{r}_V(t), \tag{V.1}$$

$$\mathbf{r}_Z(t + \Delta t) = \mathcal{P}_Z \mathbf{r}_Z(t), \tag{V.2}$$

where $\mathbf{r}_V(t)$ denotes the set of points in TSV, $\mathbf{r}_V(t) \in \mathbb{TSV}(t)$; $\mathbf{r}_Z(t)$ the set of points on a z-surface, $\mathbf{r}_Z(t) \in \mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}(t)$.

For the migration of discrete points on z-curve $\mathbf{C}_{\mathrm{z}}^{\mathrm{A,B}}$, I take a recursion approach. The initial $\mathbf{C}_{\mathrm{z}}^{\mathrm{A,B}}$ at a starting epoch is obtained by the tracing method proposed in Chapter III, or other QSIC solvers like the Levin's method [2], marching methods [11], subdivision methods [9], etc. [5][7]. Subsequent $\mathbf{C}_{\mathrm{z}}^{\mathrm{A,B}}$'s are derived through a procedure based on the previous $\mathbf{C}_{\mathrm{z}}^{\mathrm{A,B}}$. I propose such a recursive procedure by analyzing the dynamics of $\mathbb{C}_{\mathrm{z}}^{\mathrm{A,B}}$ from t to $t + \Delta t$. Challenges in the dynamic analysis are illustrated in Figure V-2. Firstly, as a part of z-surface, the z-curve moves along with z-surface, but it may no longer align with the Earth surface when the z-surface inclines at a different angle. Secondly, being constrained by $\mathbb{TSV}^{\mathrm{A,B}}$, the z-curve will be shorten or elongated due to the change of $\mathbb{TSV}^{\mathrm{A,B}}$. As will be discussed in section V.5, the proposed method adopts numerical approaches to overcome these challenges.

The remainder of this chapter is organized as follows. I first analyze the pair-satellite motion in Section V.2. Next, I derive the transformation functions $\mathcal{P}_V$ and $\mathcal{P}_Z$ for TSV and *z*-surface in Sections V.3 & V.4, respectively. In Section V.5, I propose the numeric algorithm for z-curve migration in detail. I show our experimental process, results, and analyses in Section V.7.

V.2 Paired-motion of Two Satellites

Main contributing factors to the dynamics of z-surface include the phases and velocities of the Earth's self-spinning, orbiting of ($\mathcal{A}$, $\mathcal{B}$) with different elevation angles, as well as the change of atmosphere properties. To simplify the analysis, I fix the Earth motion by using ECEF frame, and adopt the ephemeris model to represent satellite motions, where orbits are sinusoid-like trajectories.

When ($\mathcal{A}$, $\mathcal{B}$) are in paired motion, $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ is being "dragged" along in the 3D space, as shown in Figure V-3. The motion effect on $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ can be composed from translating, rotating, reflection, and scaling, which are mainly affected by the following four factors:

1) The *displacement* of $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ can be determined by the midpoint point o′ between ($\mathcal{A}$, $\mathcal{B}$):

$$\mathbf{o}' = \frac{1}{2}(\mathbf{r}^A + \mathbf{r}^B) = (\frac{x^A+x^B}{2}, \frac{y^A+y^B}{2}, \frac{z^A+z^B}{2}, 1)^T. \qquad \text{(V.3)}$$

2) The *inclination* of $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ with respect to the ECEF axes can be determined by the orientation n_z, which is defined as a unit vector in the direction from $\mathcal{A}$ to $\mathcal{B}$:

$$\mathbf{n}' = \frac{\mathbf{r}^B-\mathbf{r}^A}{2c} = (\frac{x^B-x^A}{2c}, \frac{y^B-y^A}{2c}, \frac{z^B-z^A}{2c}, 0)^T, \qquad \text{(V.4)}$$

and c is half of the foci distance, i.e. $c = \frac{1}{2}\|\mathbf{r}^B - \mathbf{r}^A\|$.

3) The *opening direction* of the hyperbolic sheet towards $\mathcal{A}$ or $\mathcal{B}$ is determined by the sign of its semi-major axis a. The value a is the difference of signal propagation errors between $\mathcal{A}$ and $\mathcal{B}$:

$$a = \frac{v}{2}\left[\left(\delta_{io}^{A} - \delta_{io}^{B}\right) + \left(\delta_{tr}^{A} - \delta_{tr}^{B}\right) - \left(\delta_{sc}^{A} - \delta_{sc}^{B}\right)\right] + \frac{k}{2}, \tag{V.5}$$

where v is the speed of light in vacuum, and δ_{io}, δ_{tr}, δ_{sc} the timing delays caused by Ionospheric and tropospheric refractions, satellite clock offset, respectively. Referred to Figure V-3, when $a > 0$, $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ is the hyperbolic sheet closer to A, called wing-A (W^{A}). Otherwise, it is the wing-B (W^{B}). When $a = 0$, $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ degenerates to a plane $H^{A,B}$.

4) The *eccentricity* of $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$, denoted as e, is defined as below. Note that the larger e is, the flatter $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ is.

$$e = \left|\frac{c}{a}\right| (a \neq 0). \tag{V.6}$$

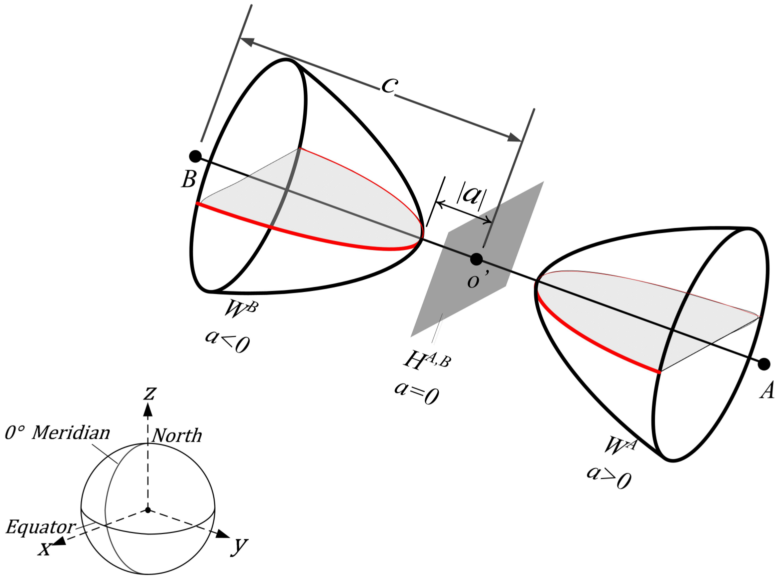


Figure V-3 an illustration of a z-surface

The two satellites may fly at different velocities and accelerations, in different orbits, elevations, and inclinations. It is exceedingly complicated to incorporate these factors in algebraic equations of closed forms. As a pragmatic alternative, I propose a discrete, time step approach, similar to the basic concept of ephemeris model, so that given positions of ($\mathcal{A}$, $\mathcal{B}$) at time t and $t + \Delta t$, I use affine transformation matrices to map a z-surface from $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}(t)$ to $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}(t + \Delta t)$. This method is inherently compatible with the computing process of ephemeris calculating and pseudorange measuring.

This work uses rotation, translation and some other transformations to describe shapes' migration. Let $\mathcal{R}_x(\omega)$, $\mathcal{R}_y(\omega)$, and $\mathcal{R}_z(\omega)$ for arbitrary positive angle ω denote the counterclockwise rotations about x-, y- and z-axes, when one is looking from a positive axis toward the origin [54]:

$$\mathcal{R}_x(\omega) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega & 0 \\ 0 & \sin\omega & \cos\omega & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{V.7}$$

$$\mathcal{R}_y(\omega) = \begin{pmatrix} \cos\omega & 0 & \sin\omega & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\omega & 0 & \cos\omega & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \tag{V.8}$$

$$\mathcal{R}_z(\omega) = \begin{pmatrix} \cos\omega & -\sin\omega & 0 & 0 \\ \sin\omega & \cos\omega & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{V.9}$$

Let $\mathcal{T}(\mathbf{v})$ for an arbitrary vector $\mathbf{v} = (x, y, z, 0)^{\boldsymbol{T}}$ represent the translation matrix [54]:

$$\mathcal{T}(\mathbf{v}) = \begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{V.10}$$

Some properties of them will be used: since $\mathcal{R}$ is a orthogonal matrix, we have $\mathcal{R}^{-1}(\omega) = \mathcal{R}^{T}(\omega) = \mathcal{R}(-\omega)$; and $\mathcal{T}^{-1}(\mathbf{v}) = \mathcal{T}(-\mathbf{v})$ [54].

V.3 TSV Migration

In this section, we firstly derive the Cartesian model of TSV in a homogenous form using 4D coordinates. I then derive the migration matrix $\mathcal{P}_V$ using homogeneous coordinates.

The $\mathbb{TSV}^{S}$ has a quadratic shape – cone as the primary component, with several visibility constraints. Let θ denote half of the opening angle (For GPS, θ=13.8° [29]), $\mathbf{r}^S$ the satellite antenna coordinate, $\mathbf{r}^P$the pointing target. For convenience, I assume $\mathbf{r}^P$ to be the origin of ECEF frame, i.e. $\mathbf{r}^P = (0,0,0,1)^T$. Let $\mathbf{d}_t$ denote the direction of radiation of S's antenna array, and $\mathbf{d}_t = \frac{\mathbf{r}^P - \mathbf{r}_t^S}{\|\mathbf{r}^P - \mathbf{r}_t^S\|} = (\frac{-x_t^S}{\|\mathbf{r}_t^S\|}, \frac{-y_t^S}{\|\mathbf{r}_t^S\|}, \frac{-z_t^S}{\|\mathbf{r}_t^S\|}, 0)^T$.

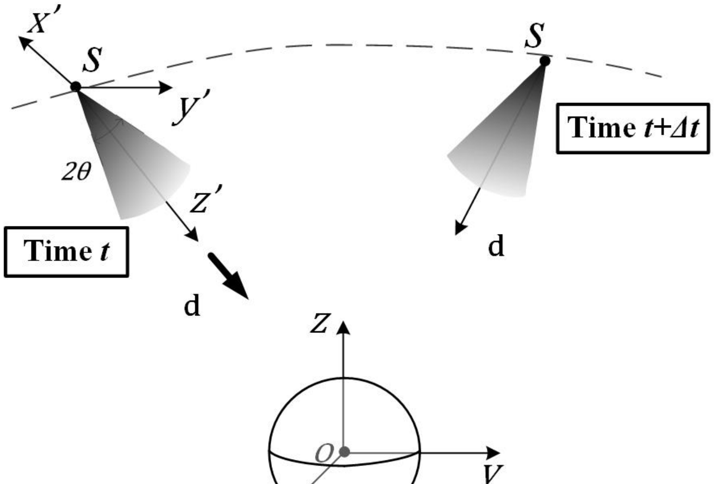


Figure V-4 Illustration of TSV Migration

Theorem V.1: The TSV of a satellite S, $\mathbb{TSV}^{\mathrm{S}}$, about the reference ellipsoid defined in WGS84 has the following implicit Cartesian expression using *homogenous coordinates*:

$$\mathbb{TSV}^{\mathrm{S}}(\mathbf{r}_t^S)\colon \begin{cases} \mathbf{r}^T Q^E \mathbf{r} \leq 0, \\ \mathbf{r}^T Q_t^S \mathbf{r} \leq 0, \\ (H_t^S)^T \mathbf{r} \leq 0, \\ (H_t^E)^T \mathbf{r} \leq 0, \end{cases} \tag{V.11}$$

where Q and H are used to denote symmetric matrix and vector of coefficients, respectively:

$$Q^E = diag\left(-\frac{1}{a_E^2}, -\frac{1}{a_E^2}, -\frac{1}{b_E^2}, -1\right),$$

$$H_t^S = \left(x_t^S, y_t^S, z_t^S, -\|\mathbf{r}_t^S\|^2\right)^T,$$

$$H_t^E = \mathbf{d}_t,$$

$$Q_t^S = \mathcal{Q}_{yxr}^T Q_\theta^{cone} \mathcal{Q}_{yxr}.$$

Q_θ^{cone} and $\mathcal{Q}_{yxr}$ are defined as below:

$$Q_\theta^{cone} = diag(1,1,-tan^2\theta, 0),$$

$$\mathcal{Q}_{yxr}(-\beta, \alpha, -\mathbf{r}_t^S) = \mathcal{R}_y(-\beta)\mathcal{R}_x(\alpha)\mathcal{T}(-\mathbf{r}_t^S) = \begin{pmatrix} \cos\beta & -\sin\alpha\sin\beta & -\cos\alpha\sin\beta & -x_t^S \\ 0 & \cos\alpha & -\sin\alpha & -y_t^S \\ \sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta & -z_t^S \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$\cos\alpha = \frac{d_{z,t}}{\sqrt{d_{y,t}^2 + d_{z,t}^2}}, \sin\alpha = \frac{d_{y,t}}{\sqrt{d_{y,t}^2 + d_{z,t}^2}}, \cos\beta = \sqrt{d_{y,t}^2 + d_{z,t}^2}, \sin\beta = d_{x,t}.$$

Proof: This system is derived from the TSV model in Chapter IV. In (V.11), the derivation of 1st, 3rd, and 4th inequalities is straightforward, and thus omitted. The 1st inequality of Q^E is a constraint to exclude points inside the Earth. a_E (6378137 meters) and b_E (≈6356752.314140 meters) are known constants that represent the equatorial radius and polar radius, respectively [18]. The 3rd inequality of H_t^S ensures all points are visible to the satellite. The 4th inequality of H_t^E excludes points back of S.

The 2nd inequality of Q_t^S describes points within a double cone centered at S pointing in $\mathbf{d}_t$. Following the same proof process in Chapter IV, Q_t^S is derived by applying transformation $\mathcal{Q}_{yxr}$ to a canonical cone $\mathbf{r}^T Q_\theta^{cone} \mathbf{r} \leq 0$, i.e. $Q_t^S = \mathcal{Q}_{yxr}^T Q_\theta^{cone} \mathcal{Q}_{yxr}$. $\mathcal{Q}_{yxr}$ is a composite transformation operation including the rotation $\mathcal{R}_y(-\beta)$ about y-axis, rotation $\mathcal{R}_x(\alpha)$ about x-axis, and translation $\mathcal{T}(-\mathbf{r}_t^S)$. Here, α is the angle from the projection of $\mathbf{d}_t$ on the yz-plane to the z-axis, and β the angle from $\mathbf{d}_t$ to this projection. ■

To transform discrete points generated from Theorem V.1, i.e. $\mathbf{r}_V(\mathbf{r}_t^S) \in \mathbb{TSV}^{\mathrm{S}}(\mathbf{r}_t^S)$, into the pose at time $t + \Delta t$, i.e. $\mathbf{r}_V\left(\mathbf{r}_{t+\Delta t}^S\right) \in \mathbb{TSV}^{\mathrm{S}}(\mathbf{r}_{t+\Delta t}^S)$, one can adjust the cone's pose to match the new direction of radiation by rotation, and then translate them to match the cone's apex to sit on the new position of S.

Mathematically, let discrete points multiply $\mathcal{P}_V = \mathcal{T}(\Delta \mathbf{r}_t^S)\mathcal{R}$, where $\mathcal{T}$ is referred to (10), Δ refers to a difference operator, i.e. $\Delta \mathbf{r}_t^S = \mathbf{r}_{t+\Delta t}^S - \mathbf{r}_t^S$. $\mathcal{R}$ is a rotation from the vector $\mathbf{d}_t$ to $\mathbf{d}_{t+\Delta t}$, which can be obtained by the Rodrigue's formula [103]:

$$\mathcal{R}_d = I + \sin\phi\,[U]_\times + (1 - \cos\phi)[U]_\times^2, \tag{V.12}$$

where I is a 3D identity matrix, U a vector equal to the unit cross product of $[\mathbf{d}_t]_3$ and $[\mathbf{d}_{t+\Delta t}]_3$ (the sub-vector of first three elements of $\mathbf{d}_t$ and $\mathbf{d}_{t+\Delta t}$), ϕ the angle between $[\mathbf{d}_t]_3$ and $[\mathbf{d}_{t+\Delta t}]_3$, and $[\cdot]_\times$ the skew-symmetric matrix of a vector [68].

$$U = \frac{[\mathbf{d}_t]_3 \times [\mathbf{d}_{t+\Delta t}]_3}{\|[\mathbf{d}_t]_3 \times [\mathbf{d}_{t+\Delta t}]_3\|},\ \phi = cos^{-1}(\mathbf{d}_t \cdot \mathbf{d}_{t+\Delta t}), \tag{V.13}$$

where “×” and “·” refers to cross-product and dot-product. We are assuming $\phi \neq 0$ or π and thus $\|[\mathbf{d}_t]_3 \times [\mathbf{d}_{t+\Delta t}]_3\| \neq 0$, because otherwise that implies the satellite is moving very slowly ($\mathbf{d}_t = \mathbf{d}_{t+\Delta t}$) or has flew to the opposite of the orbit ($\mathbf{d}_t = -\mathbf{d}_{t+\Delta t}$) during Δt. As such, $\mathcal{P}_V$ is derived as follows:

$$\mathcal{P}_V = \mathcal{T}(\Delta \mathbf{r}_t^S)\mathcal{R} = \begin{pmatrix} \mathcal{R}_d & \begin{matrix} \Delta x_t^S \\ \Delta y_t^S \\ \Delta z_t^S \end{matrix} \\ \mathbf{0} & 1 \end{pmatrix}. \tag{V.14}$$

V.4 Z-Surface Migration

In this section, I first rewrite the z-surface Cartesian model into the homogenous form. Then we derive its migration matrix $\mathcal{P}_Z$.

Based on Chapter III, the z-surface $\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}}$ can be modeled as a hyperbolic sheet $\mathfrak{S}_z^{A,B}$, constrained by the TSVs of ($\mathcal{A}$, $\mathcal{B}$), or:

$$\mathbb{S}_{\mathrm{z}}^{\mathrm{A,B}} = \mathfrak{S}_z^{A,B} \cap \mathbb{TSV}^{\mathrm{A}} \cap \mathbb{TSV}^{\mathrm{B}}. \tag{V.15}$$

Theorem V.2: The z-surface hyperbolic sheet $\mathfrak{S}_z^{A,B}$ has the following implicit Cartesian expression using homogenous coordinates:

$$\mathfrak{S}_z^{A,B} \colon \begin{cases} \mathbf{r}^T Q^E \mathbf{r} \leq 0, \\ \mathbf{r}^T Q_t^Z \mathbf{r} = 0, \\ a_t^k (H_t^Z)^T \mathbf{r} \geq 0, \end{cases} \tag{V.16}$$

where the coefficients Q_t^Z and H_t^Z are listed as below:

$$Q_t^Z = \begin{pmatrix} C^2 - (4a_t)^2 & CD & CE & GC + (4a_t)^2 x_t^A \\ CD & D^2 - (4a_t)^2 & DE & GD + (4a_t)^2 y_t^A \\ CE & DE & E^2 - (4a_t)^2 & GE + (4a_t)^2 z_t^A \\ GC + (4a_t)^2 x_t^A & GD + (4a_t)^2 y_t^A & GE + (4a_t)^2 z_t^A & G^2 - (4a_t)^2 \|\mathbf{x}_t^A\|^2 \end{pmatrix},$$

$$H_t^Z = (C, D, E, F)^T,$$

$C = 2(x_t^A - x_t^B), D = 2(y_t^A - y_t^B), E = 2(z_t^A - z_t^B), F = \|\mathbf{x}_t^B\|^2 - \|\mathbf{x}_t^A\|^2, G = F - (2a_t)^2.$

Proof: This theorem is mathematically derived from in Chapter III. The 1st inequality of Q^E limits the points outside of the Earth. The 2nd inequality of Q_t^Zrequires points be on the hyperboloid of two-sheet. Q_t^Z is derived by expanding coefficients in Theorem III.2 and then re-organizing up. I leave this process to readers. The 3rd inequality filters one of the sheets out according the sign of the semi-major axis a_t^k (see (V.5)). ■

The migration of z-surface $\mathbb{S}_z^{\mathrm{A,B}}$ consists of following components: a *translation* to move the displacement (V.3), a *rotation* to adjust the inclination (V.4), a *scaling* and a *reflection* to tune the opening angle and direction, respectively.

The executing order, however, needs to be considered carefully. The scaling, especially, needs to be uniformly applied in the shape that is aligned with the origin and axes of the ECEF frame. As such, in order to get $\mathfrak{S}_{t+\Delta t}^{A,B}$ from $\mathfrak{S}_t^{A,B}$, one must firstly align the pose of $\mathfrak{S}_t^{A,B}$ downward with the ECEF frame before scale it and reflect it, and finally adjust upward to be the pose of $\mathfrak{S}_{t+\Delta t}^{A,B}$. The overall process of transformations is listed as below, where $\mathbf{r}_t \in \mathfrak{S}_t^{A,B}$, $\mathbf{r}_{t+\Delta t} \in \mathfrak{S}_{t+\Delta t}^{A,B}$ are points in source and target shapes, and $\mathbf{r}_{1,2,3,4,5}$ are points in intermediate shapes.

<u>Step 1</u>: $\mathbf{r}_1 = \mathcal{T}(-\mathbf{o}_t')\mathbf{r}_t$: Translate the center of $\mathfrak{S}_t^{A,B}$ from the midpoint $\mathbf{o}_t'$ of ($\mathcal{A}$, $\mathcal{B}$) to the origin of ECEF frame.

<u>Step 2</u>: $\mathbf{r}_2 = \mathcal{R}_x(-\alpha_t)\mathcal{R}_z(\gamma_t)\mathbf{r}_1$: Rotate $\mathbf{r}_1$'s γ_t and α_t angles about z- and x-axes, respectively, so that the foci of the hyperbolic sheet, or ($\mathcal{A}$, $\mathcal{B}$), are placed on the y-axis at c and – c away from the origin. Here γ_t is the angle from $\mathbf{n}_t'$'s projection in the xy-plane (see

$(n'_{t,x}, n'_{t,y}, 0)^T$ in Figure V-5),to the y axis, while α_t is the angle from the rotated vector $(0, \sqrt{n'^2_{t,x} + n'^2_{t,y}}, n'_{t,z})^T$ to the yz-plane and the y-axis [54]:

$$\cos\gamma_t = \frac{n'_{y,t}}{\sqrt{n'^2_{x,t}+n'^2_{y,t}}}, \sin\gamma_t = \frac{n'_{x,t}}{\sqrt{n'^2_{x,t}+n'^2_{y,t}}}; \tag{V.17}$$

$$\cos(\alpha_t) = \sqrt{n'^2_{x,t} + n'^2_{y,t}}, \sin(\alpha_t) = n'_{z,t}. \tag{V.18}$$

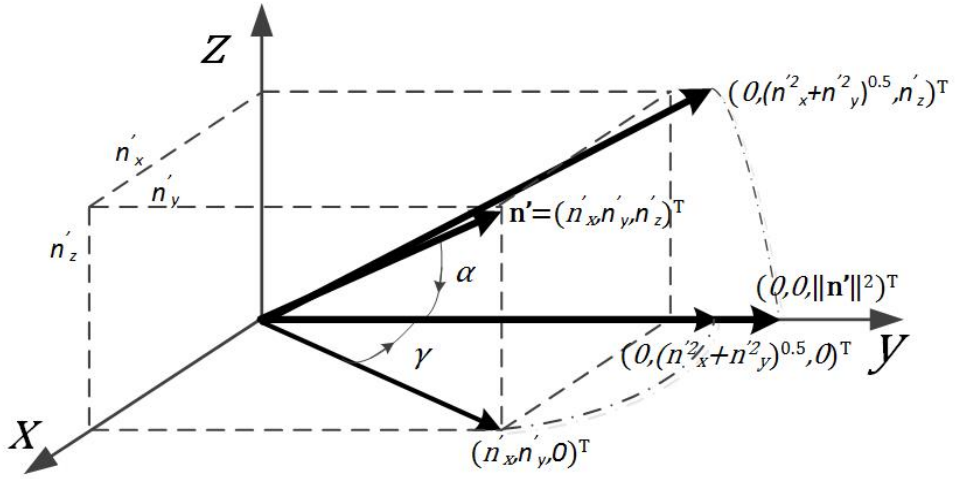

Z
$(0,(n'^2_x+n'^2_y)^{0.5},n'_z)^T$
n'_x
n'_y
$\mathbf{n}'=(n'_x,n'_y,n'_z)^T$
n'_z
α
$(0,0,\|\mathbf{n}'\|^2)^T$
γ
$(0,(n'^2_x+n'^2_y)^{0.5},0)^T$
Y
X
$(n'_x,n'_y,0)^T$

Figure V-5 Rotation angles

After this step, the intermediate hyperbolic sheet can be written in a canonical form as follows [53]:

$$\frac{x^2}{a_t^2} - \frac{y^2}{c_t^2} + \frac{z^2}{a_t^2} = -1. \ (a_t y > 0), \tag{V.19}$$

where $a_t y > 0$ ensures the correct polarity of the hyperbolic sheet.

<u>Step 3</u>: $\mathbf{r}_3 = \mathcal{S}\mathbf{r}_2$: Scale the hyperbolic sheet to be:

$$\frac{x^2}{a_{t+\Delta t}^2} - \frac{y^2}{c_{t+\Delta t}^2} + \frac{z^2}{a_{t+\Delta t}^2} = -1. \ (a_t y > 0), \tag{V.20}$$

where the matrix $\mathcal{S}$ is derived by comparing (V.20) with (V.19):

$$\mathcal{S} = diag\left(\frac{|a_t|}{|a_{t+\Delta t}|}, \frac{|c_t|}{|c_{t+\Delta t}|}, \frac{|a_t|}{|a_{t+\Delta t}|}, 1\right). \tag{V.21}$$

Step 4: $\mathbf{r}_4 = \mathcal{F}\mathbf{r}_3$: Reflect the opening direction of the hyperbolic sheet to ensure $a_{t+\Delta t}y > 0$. If $a_{t+\Delta t}a_t > 0$, this step can be skipped, since the opening direction does not change. If $a_{t+\Delta t}a_t < 0$, i.e. $a_{t+\Delta t}y < 0$, all points $\mathbf{r}_3$'s in (V.20) need to be reflected with along the y-axis [54]:

$$\mathcal{F} = diag(1, sign(a_{t+\Delta t}a_t), 1,1), \tag{V.22}$$

where $sign(x) = 1$ if $x > 0$, or -1 if $x < 0$. If $a_{t+\Delta t}a_t > 0$, $\mathcal{F}$ becomes an identity matrix.

Step 5: $\mathbf{r}_5 = \mathcal{R}_z(-\gamma_{t+\Delta t})\mathcal{R}_x(\alpha_{t+\Delta t})\mathbf{r}_4$: Adjust the shape $\mathbf{r}_4$'s orientation to match that of $\mathbf{r}_{t+\Delta t} \in \mathfrak{S}_{t+\Delta t}^{A,B}$. Here $\alpha_{t+\Delta t}$ and $\gamma_{t+\Delta t}$ are similar angles of $\mathbf{n}'_{t+\Delta t}$ to those in the 2nd step [54]:

$$\cos\gamma_{t+\Delta t} = \frac{n'_{y,t+\Delta t}}{\sqrt{n'^2_{x,t+\Delta t}+n'^2_{y,t+\Delta t}}}, \sin(\gamma_{t+\Delta t}) = \frac{n'_{x,t+\Delta t}}{\sqrt{n'^2_{x,t+\Delta t}+n'^2_{y,t+\Delta t}}}; \tag{V.23}$$

$$\cos\alpha_{t+\Delta t} = \sqrt{n'^2_{x,t+\Delta t} + n'^2_{y,t+\Delta t}}, \sin\alpha_{t+\Delta t} = n'_{z,t+\Delta t}. \tag{V.24}$$

Both angles are negated to have a counter-clockwise rotation, because at this time, $\mathbf{n}'_{t+\Delta t}$ is the target orientation of rotation while in the 2nd step, $\mathbf{n}'_t$ is the original orientation of rotation.

Step 6: $\mathbf{r}_{t+\Delta t} = \mathcal{T}(\mathbf{o}'_{t+\Delta t})\mathbf{r}_5$: Translate the shape from the origin to the midpoint of ($\mathcal{A}$, $\mathcal{B}$) at $\boldsymbol{t} + \Delta\boldsymbol{t}$, to get final pose.

To sum up, the transformation $\mathcal{P}_Z$ in $\mathbf{r}_{t+\Delta t} = \mathcal{P}_Z\mathbf{r}_t$ is:

$$\mathcal{P}_Z = \mathcal{T}(\mathbf{o}'_{t+\Delta t})\mathcal{R}_z(-\gamma_{t+\Delta t})\mathcal{R}_x(\alpha_{t+\Delta t})\mathcal{F}\mathcal{S}\mathcal{R}_x(-\alpha_t)\mathcal{R}_z(\gamma_t)\mathcal{T}(-\mathbf{o}'_t), \tag{V.25}$$

To simplify this representation, we can group $\mathcal{R}_x(\omega)\mathcal{R}_z(\theta)$, $\mathcal{T}(\mathbf{r})$ together as $\mathcal{Q}_{xzr}(\alpha, \gamma, \mathbf{r})$, which can be computed as:

$$\mathcal{Q}_{xzr}(\alpha, \gamma, \mathbf{r}) = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 & x \\ \cos\alpha\sin\gamma & \cos\gamma & -\sin\alpha & y \\ \sin\alpha\sin\gamma & \cos\gamma & \cos\alpha & z \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad \text{(V.26)}$$

In $\mathcal{Q}_{xzr}$, the order of $\mathcal{T}$ and $\mathcal{R}$ does not matter, but $\mathcal{R}_x$ must be in front of $\mathcal{R}_z$. Since $\mathcal{Q}_{xzr}^{-1}(\alpha, \gamma, \mathbf{r}) = (\mathcal{R}_x(\alpha)\mathcal{R}_z(\gamma)\mathcal{T}(\mathbf{r}))^{-1} = \mathcal{T}^{-1}(\mathbf{r})\mathcal{R}_z^{-1}(\gamma)\mathcal{R}_x^{-1}(\alpha) = \mathcal{T}(-\mathbf{r})\mathcal{R}_x(-\alpha)\mathcal{R}_z(-\gamma)$, the final transformation matrix $\mathcal{P}_Z$ can also be given as follows:

$$\mathcal{P}_Z = \mathcal{Q}_{xzr}^{-1}(-\alpha_{t+\Delta t}, \gamma_{t+\Delta t}, -\mathbf{o}'_{t+\Delta t})\mathcal{F}\mathcal{S}\mathcal{Q}_{xzr}(-\alpha_t, \gamma_t, -\mathbf{o}'_t). \quad \text{(V.27)}$$

V.5 Z-Curve Migration

A z-curve $\mathbb{C}^{\mathrm{A,B}}$ is a QSIC of z-surface (V.16) and the Earth surface, or formally:

$$\mathbb{C}^{\mathrm{A,B}} = \mathbb{S}^{\mathrm{A,B}} \cap \mathbb{E} = \mathfrak{S}^{A,B} \cap \mathbb{TSV}^{\mathrm{A}} \cap \mathbb{TSV}^{\mathrm{B}} \cap \mathbb{E}. \quad \text{(V.28)}$$

The main focus in this section is the migration of z-curve's discrete points: *given* $\mathbf{r}_{\mathrm{t}} \in \mathbf{C}_{\mathrm{t}}^{\mathrm{A,B}}$, *what is its relation with* $\mathbf{r}_{\mathrm{t+\Delta t}} \in \mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$*?* How to obtain the input $\mathbf{r}_t \in \mathbf{C}_{\mathrm{t}}^{\mathrm{A,B}}$ will not be discussed here, since it is a classic QSIC tracing problem, and our previous work [2] presents an analytic QSIC solver for it. Its migration to $\mathbf{r}_{t+\Delta t} \in \mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$ can be accomplished by the numerical algorithm present in this section.

Overview

As illustrated in Figure V-2, the bulk of $\mathbf{C}^{\mathrm{A,B}}$ behavior is migrating along with the z-surface $\mathbb{S}^{\mathrm{A,B}}$. On one hand, as $\mathbb{S}^{\mathrm{A,B}}$ becomes steeper or flatter at $t{+}\Delta t$, points in $\mathbf{C}^{\mathrm{A,B}}$ may be lifted up above ground or brought down under ground. They shall be leveled downward and upward to align with Earth surface, respectively. In the meantime, the uniformness of intervals shall be maintained to some degree. On the other hand, the z-curve varies in length due to the change of $\mathbb{TSV}^{\mathrm{A,B}}$. When the coverage zone shrinks, those out-of-coverage points need to be filtered out of $\mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$. When $\mathbb{TSV}^{\mathrm{A,B}}$ enlarges, more discrete points on $\mathbb{C}^{\mathrm{A,B}}$ shall be inserted in the front or appended in the end of $\mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$. The uniformness shall also be retained as that of $\mathbf{C}_{\mathrm{t}}^{\mathrm{A,B}}$.

The proposed method overcomes these challenges. Its three general steps are listed as follows, where $\mathbf{r}_t \in \mathbf{C}_{\mathrm{t}}^{\mathrm{A,B}}$ is the input, $\mathbf{r}_{t+\Delta \mathrm{t}} \in \mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$ the output, and $\mathbf{p} \in \mathbf{P}$, $\mathbf{q} \in \mathbf{Q}$ intermediate shapes.

<u>*Step 1*</u>: Affine transformation by $\mathcal{P}_Z$

Moving along the z-surface is mathematically achieved by multiplying $\mathbf{r}_t \in \mathbf{C}_{\mathrm{t}}^{\mathrm{A,B}}$ with the same motion matrix $\mathcal{P}_Z$:

$$\mathbf{p} = \mathcal{P}_Z \mathbf{r}_t, \tag{V.29}$$

I will not discuss this step in detail hereafter.

<u>*Step* 2</u>: Earth surface alignment

I then apply the Newton's method on each $\mathbf{p}$ to iteratively solve for a nearby point $\mathbf{q}$ ($\in$ $\mathbf{Q}$) on the hyperbolic sheet's intersection with Earth, i.e. $\mathfrak{S}_{t+\Delta t}^{A,B} \cap \mathbb{E}$. This step is elaborated in next section.

<u>*Step* 3</u>: Boundary points alignment

Out-of-coverage $\mathbf{q}$'s shall be trimmed out of $\mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$, whereas more in-coverage $\mathbf{q}$'s need be collected. Due to the nature of this problem, I design a numerically marching based algorithm, which progressively collects discrete points along the z-curve until certain stopping conditions are met. It will be detailed later.

Earth Surface Alignment

The input of this step is $\mathbf{p} \in \mathbf{P} \subseteq \mathbb{S}_{t+\Delta t}^{\mathrm{A,B}}$, a set of points on z-surface and near the Earth surface. The output of this step is: for every $\mathbf{p}$, find a point $\mathbf{q}$ nearby on the Earth, such that $\mathbf{q} \in \mathbf{Q} \subseteq \mathfrak{S}_{t+\Delta t}^{A,B} \cap \mathbb{E}$. As it is hard to find a closed-form solution, we formulate the problem numerically:

Given an initial point $\mathbf{r}^{(0)} = (x^{(0)}, y^{(0)}, z^{(0)}, 1)^T$ *such that* ${\mathbf{r}^{(0)}}^T Q_{t+\Delta t}^Z \mathbf{r}^{(0)} = 0$*, find a point* $\mathbf{r}^{(m)} = (x^{(m)}, y^{(m)}, z^{(m)}, 1)^T$ *with at most* m *iterations, such that* ${\mathbf{r}^{(m)}}^T Q_{t+\Delta t}^Z \mathbf{r}^{(m)} = 0$ *and* ${\mathbf{r}^{(m)}}^T Q^E \mathbf{r}^{(m)} = 0$.

Here the initial value $\mathbf{r}^{(0)}$ refers to each $\mathbf{p}$ point and the final $\mathbf{r}^{(m)}$ is a corresponding $\mathbf{q}$ point. Being a root-finding problem, this problem is underdetermined - there are only two equations $Q_{t+\Delta t}^Z$ and Q^E for three variables x, y, z. The root searching starts from $\mathbf{p}$, but may head in an infinite number of directions towards the Earth surface.

A third condition is needed, to guide the root searching process. Moreover, I may utilize it to retain the uniformness of $\mathbf{q} \in \mathbf{Q}$. The uniformness of $\mathbf{C}_{\mathrm{t}}^{\mathrm{A,B}}$ is controlled in Chapter III, by the uniformly separated geocentric latitude φ_i ($\varphi_{i+1} - \varphi_i$ is constant). I thus need to guide the root searching to proceed along the latitude-surface that each $\mathbf{p}$ is on. Specifically, for each $\mathbf{p}$ ($\mathbf{r}^{(0)}$), whose geocentric coordinate is $(\lambda^{(0)}, \varphi^{(0)})$, let $\Phi^{(0)}$ denote the surface in space with geocentric latitude $\varphi^{(0)}$. As shown in Figure V-6, $\Phi^{(0)}$ is a cone sitting on the origin with $\varphi^{(0)}$ inclination angle from the equator plane. $\Phi^{(0)}$ can be derived using [69]:

$$\begin{cases} x = d\cos(\lambda)\cos(\varphi), \\ y = d\sin(\lambda)\cos(\varphi), \\ z = d\sin(\varphi), \end{cases} \tag{V.30}$$

with $d = \sqrt{x^2 + y^2 + z^2}$. $\Phi^{(0)}$ surface has the form of $z = \sqrt{x^2 + y^2 + z^2}\,\sin(\varphi^{(0)})$ by fixing φ to be $\varphi^{(0)}$.

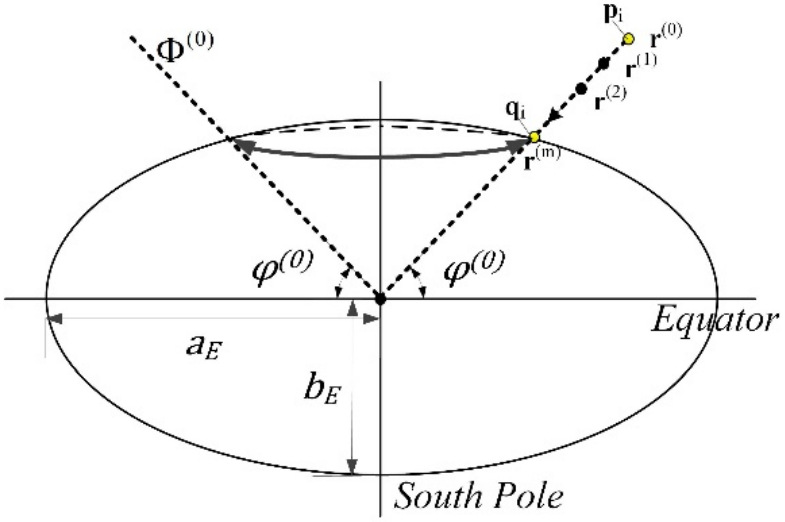


Figure V-6 Root searching direction and geocentric coordinate

By adding the third condition, I rephrase the problem into:

Given an initial value $\mathbf{r}^{(0)}$ *(geocentric latitude is* $\varphi^{(0)}$*), solve for a root of the system of equations* $\mathbf{F} = [f_E(x,y,z) = 0,\ \mathrm{f_Z(x,y,z)} = 0, \mathrm{f_\varphi(x,y,z)} = 0]$, where $\mathrm{f_E}$, $\mathrm{f_Z}$, $\mathrm{f_\varphi}$ are:

$$\mathrm{f_E}(x,y,z) = \mathbf{r}^\mathrm{T}\mathrm{Q}^\mathrm{E}\mathbf{r},$$

$$\mathrm{f_Z}(x,y,z) = d^B - d^A - 2a,$$

$$\mathrm{f_\varphi}(x,y,z) = z - \sqrt{x^2 + y^2 + z^2}\,sin(\varphi^{(0)}),$$

and $d^A = \sqrt{(x - x^A)^2 + (y - y^A)^2 + (z - z^A)^2}$, $d^B = \sqrt{(x - x^B)^2 + (y - y^B)^2 + (z - z^B)^2}$.

I solve it by the Newton's method. It takes advantage of the derivative of **F** (Jacobian Matrix J), which is used to iteratively update the intermediate point $\mathbf{r}^{(\mathrm{i})}$ in a way that can decrease its evaluation value of the system, i.e. $\left|\mathbf{F}\left(\mathbf{r}^{(\mathrm{i})}\right)\right| < \left|\mathbf{F}\left(\mathbf{r}^{(\mathrm{i}-1)}\right)\right|$, until which reaches to zero [70].

$$\mathrm{J}(x,y,z) = \begin{pmatrix} -\frac{2x}{a_E^2} & -\frac{2y}{a_E^2} & -\frac{2z}{b_E^2} \\ \frac{x-x^B}{d^B} - \frac{x-x^A}{d^A} & \frac{y-y^B}{d^B} - \frac{y-y^A}{d^A} & \frac{z-z^B}{d^B} - \frac{z-z^A}{d^A} \\ \frac{-x\,sin(\varphi^{(0)})}{\sqrt{x^2+y^2+z^2}} & \frac{-y\,sin(\varphi^{(0)})}{\sqrt{x^2+y^2+z^2}} & 1 - \frac{z\,sin(\varphi^{(0)})}{\sqrt{x^2+y^2+z^2}} \end{pmatrix} \tag{V.31}$$

The procedure to solve nonlinear systems of equations is given in. The iteration may not converge, when the hyperbolic sheet does not intersect the Earth at the specific latitude-surface, i.e. $\mathfrak{S}^{\mathrm{A,B}} \cap \mathbb{E} \cap \Phi^{(0)} = \emptyset$. In this special case, the procedure shall output void.

Table V-1 Algorithm: Pseudocode of Earth Alignment Using Newton's method

Input: $x^{(0)},\ y^{(0)}, z^{(0)}, m$. **Output**: $x^{(m)}, y^{(m)}, z^{(m)}$

Method:

```
1:  set $i$ = 1.
2:  if $i$ < $m$, repeat:
3:      calculate $\mathbf{F}$ and $J$ using $x^{(i-1)}, y^{(i-1)}, z^{(i-1)}$
4:      solve the 3×3 linear system $J(\Delta x, \Delta y, \Delta z) = \mathbf{F}$
5:      update $(x^{(i)}, y^{(i)}, z^{(i)})$=$(x^{(i-1)}, y^{(i-1)}, z^{(i-1)})$-$(\Delta x, \Delta y, \Delta z)$
6:      if $\sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$ < a small threshold:
7:          stop and return $x^{(i)}, y^{(i)}, z^{(i)}$
8:      update $i$++
9:  if $\sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$ < a small threshold:
10:     return $x^{(m)}, y^{(m)}, z^{(m)}$
11: else return void
```

For actual implementation, I suggest several points of concerns. First, I actually use a modified Newton's method to improve the typical quadratic convergence rate [70] to a cubic convergence, to save iteration steps. Due to page limit, we direct the readers to Homeier's Theorem 2 in [73] for a better version of lines 4-5 in Second, this system $\mathbf{F} = \mathbf{0}$ may have two solutions when the z-curve intersects twice with the latitude circle. One can guide the iterations converge to the correct region under satellite's coverage by limiting the step size (e.g. less than a few kilometers), as the correct local optimum should be close to the initial point. Third, the system $\mathbf{F} = \mathbf{0}$ may be ill conditioned, as its coefficients are not normalized, which increases the difficulty of convergence. One can normalize each sub-function in **F** by dividing its coefficients with some common norm of all sub-functions, and adjust the Jacobian matrix accordingly.

Boundary Points Alignment

Inputs of this step are $\mathbf{q} \in \mathbf{Q}$, a set of points on the hyperbolic sheet $\mathfrak{S}_{t+\Delta t}^{A,B}$ and the Earth surface $\mathbb{E}$. Outputs are our ultimate goal - $\mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$. The marching process begins with some seed point(s). For the case of $\mathbf{Q}= \emptyset$, one needs to rerun the QSIC tracing solver in Chapter III to generate $\mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$. I adopt a *Parametric–Implicit Surface Intersection (PISI)* solver [11] to drive this marching process, which should terminate at the edge of coverage.

Let $\mathbf{S}(\lambda,\varphi) = (x(\lambda,\varphi), y(\lambda,\varphi), z(\lambda,\varphi))$ denote the parametric Earth's surface, and $f_Z(x,y,z) = 0$ denote the implicit form of the z-surface, or specifically:

$$\mathbf{S}(\lambda,\varphi) = (a_E\cos(\lambda)\cos(\varphi)\,, a_E\sin(\lambda)\cos(\varphi)\,, b_E\sin(\varphi))^{\mathrm{T}}, \tag{V.32}$$

$$f_Z(x,y,z) = d^B - d^A - 2a, \tag{V.33}$$

with geocentric latitude $\varphi \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, and longitude $\lambda \in [-\pi,\pi)$. This PISI solver is based on the observation that the tangent vector T at any point on the intersection curve $\mathbf{S} \cap (f_Z = 0)$ is orthogonal to the normal vector N_Z of the $f_Z = 0$ surface. T and N_Z are given as [11]:

$$T(s) = \mathbf{S}_\lambda \frac{d\lambda}{ds} + \mathbf{S}_\varphi \frac{d\varphi}{ds},$$

$$\mathbf{S}_\lambda = (-a_E\sin(\lambda)\cos(\varphi)\,, a_E\cos(\lambda)\cos(\varphi)\,, 0\,)^T,$$

$$\mathbf{S}_\varphi = (-a_E\cos(\lambda)\sin(\varphi)\,, -a_E\sin(\lambda)\sin(\varphi)\,, b_E\cos(\varphi))^T,$$

$$N_Z = (\frac{x-x^B}{d^B} - \frac{x-x^A}{d^A}, \frac{y-y^B}{d^B} - \frac{y-y^A}{d^A}, \frac{z-z^B}{d^B} - \frac{z-z^A}{d^A})^T,$$

where s is a unit arc-length of the intersection curve. This orthogonality, together with the *first-fundamental-form* definition of a unit arc-length s [71], make up a system of two differential equations with the unknowns as the derivatives of λ, φ on s. Specifically, by plugging (V.32) and (V.33) into the saying differential equations in [11], we can express the system after some derivations as follows:

$$\frac{d\lambda}{ds} = \pm\frac{\beta}{\Delta}, \frac{d\varphi}{ds} = \mp\frac{\alpha}{\Delta}, \tag{V.34}$$

where the sign decides the two opposite marching directions on the intersection curve, and $\Delta = \sqrt{E\beta^2 - 2F\alpha\beta + G\alpha^2}$, with E, F, G being the first-fundamental-form coefficients:

$$\alpha = \mathbf{S}_\lambda^{\mathrm{T}} N_Z, \beta = \mathbf{S}_\varphi^{\mathrm{T}} N_Z, \tag{V.35}$$

$$E = \mathbf{S}_\lambda^{\mathrm{T}} \mathbf{S}_\lambda, F = \mathbf{S}_\lambda^{\mathrm{T}} \mathbf{S}_\varphi, G = \mathbf{S}_\varphi^{\mathrm{T}} \mathbf{S}_\varphi. \tag{V.36}$$

The system (V.34), together with a starting point of marching (denoted as $(\lambda^{(0)}, \varphi^{(0)})$), is a standard initial-value problem of ordinary differential equations [70]. As such, we use a classic 4th-order Runge-Kutta (RK4) method to solve it [70]. In the RK4 solver, the step size h is the same unit of the arc-length s, which can be set approximately as the great circle distance [72] between two points on the previous z-curve $\mathbf{C}_{\mathrm{t}}^{\mathrm{A,B}}$. Let u, v denote the functions in (34), the marching solver can be designed as the Table II-2, where the **StopOrNot** function is left to readers to implement based on coverage.

Table V-2 Algorithm: Pseudocode of the Marching Solver Based on RK4

Inputs: $\lambda^{(0)},\ \varphi^{(0)}, \frac{d\lambda}{ds} = u(\lambda, \varphi), \frac{d\varphi}{ds} = v(\lambda, \varphi)$.

Outputs: new points in the unexplored area to be added to $\mathbf{C}_{\mathrm{t+\Delta t}}^{\mathrm{A,B}}$

```
set the step size $h$, set $i$=0
while (true):
    compute $\begin{cases} U_1 = u\left(\lambda^{(i)}, \varphi^{(i)}\right)h, \\ V_1 = v\left(\lambda^{(i)}, \varphi^{(i)}\right)h. \end{cases}$
    compute $\begin{cases} U_2 = u\left(\lambda^{(i)} + \frac{U_1}{2}, \varphi^{(i)} + \frac{V_1}{2}\right) h, \\ V_2 = v\left(\lambda^{(i)} + \frac{U_1}{2}, \varphi^{(i)} + \frac{V_1}{2}\right) h. \end{cases}$
    compute $\begin{cases} U_3 = u\left(\lambda^{(i)} + \frac{U_2}{2}, \varphi^{(i)} + \frac{V_2}{2}\right) h, \\ V_3 = v\left(\lambda^{(i)} + \frac{U_2}{2}, \varphi^{(i)} + \frac{V_2}{2}\right) h. \end{cases}$
    compute $\begin{cases} U_4 = u\left(\lambda^{(i)} + U_3, \varphi^{(i)} + V_3\right) h, \\ V_4 = v\left(\lambda^{(i)} + U_3, \varphi^{(i)} + V_3\right) h. \end{cases}$
    compute $\begin{cases} \lambda^{(i+1)} = \lambda^{(i)} + \frac{U_1 + 2U_2 + 2U_3 + U_4}{6}, \\ \varphi^{(i+1)} = \varphi^{(i)} + \frac{V_1 + 2V_2 + 2V_3 + V_4}{6}. \end{cases}$
    if **StopOrNot(**$\lambda^{(i+1)},\ \varphi^{(i+1)}$**)** returns true:
        stop iteration, discard ($\lambda^{(i+1)},\ \varphi^{(i+1)}$)
    save ($\lambda^{(i+1)},\ \varphi^{(i+1)}$),
    update $i = i + 1$
```

V.6 Experimental Validation & Analysis

Software platform

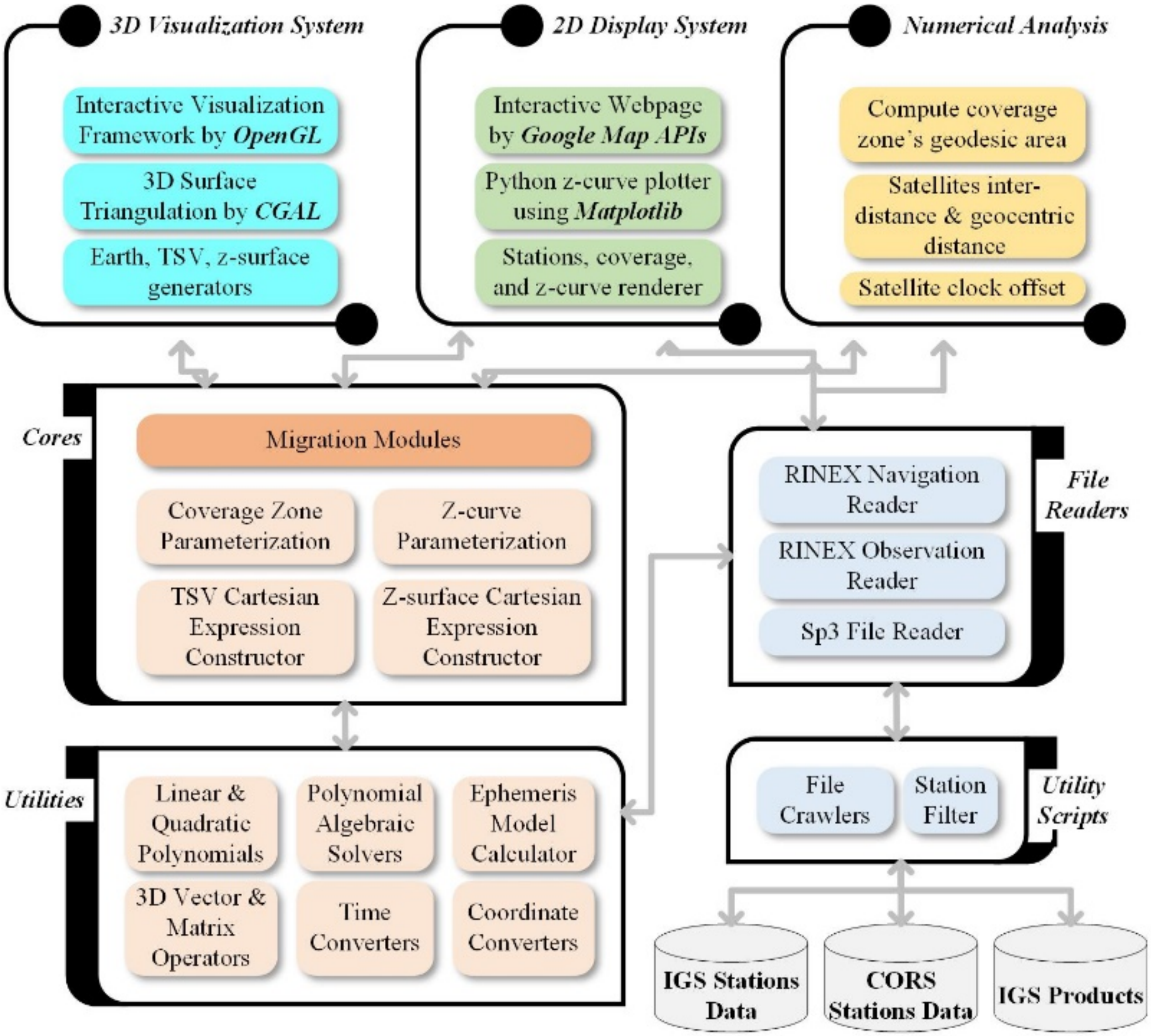

3D Visualization System
Interactive Visualization Framework by OpenGL
3D Surface Triangulation by CGAL
Earth, TSV, z-surface generators
2D Display System
Interactive Webpage by Google Map APIs
Python z-curve plotter using Matplotlib
Stations, coverage, and z-curve renderer
Numerical Analysis
Compute coverage zone's geodesic area
Satellites inter-distance & geocentric distance
Satellite clock offset
Cores
Migration Modules
Coverage Zone Parameterization
Z-curve Parameterization
TSV Cartesian Expression Constructor
Z-surface Cartesian Expression Constructor
File Readers
RINEX Navigation Reader
RINEX Observation Reader
Sp3 File Reader
Utilities
Linear & Quadratic Polynomials
Polynomial Algebraic Solvers
Ephemeris Model Calculator
3D Vector & Matrix Operators
Time Converters
Coordinate Converters
Utility Scripts
File Crawlers
Station Filter
IGS Stations Data
CORS Stations Data
IGS Products

Figure V-7 Experimental platform on static and dynamic z-curve theories

I develop a software library in C++ and Javascript for the validation, visualization, and analysis of static characteristics and dynamic behaviors of z-surface, z-curve, and satellite coverage. As outlined in Figure V-7, the library obtains measurement data from almost 4000 GNSS reference stations in the global IGS and CORS networks [23][86], to serve as ground truth. The library gets the satellite ephemeris files from NASA CDDIS server [87] to compute precise satellite positions with 1 meter or less error. The library can then construct the implicit Cartesian models, migration functions, and other utilities like time and coordinate conversions, polynomial and matrix solvers. The library supports 3D rendering via OpenGL and CGAL libraries [88][89].

Validation and a potential application

I focus to validate the migration of z-curve, because it involves the migration of z-surface as well. In Chapter III, I have validated that at the time when a predicted z-curve sweeps by, corresponding GNSS receivers will report in a timely manner, the change of order of two pseudoranges. Those who do not report the crossing are found to be anomalous, i.e. injected by errors. In other words, the algebraic QSIC solver for the z-curve proposed in Chapter III is correct and consistent with real measurements. Therefore, my target here is to match the migrated z-curve with the z-curve computed using QSIC solver in Chapter III.

The validation result is shown in Figure V-8, where two black lines are the z-curves computed using the solver in Chapter III.5, from two GPS satellites (PRN 6 and PRN 15) at TOW (Time of Week) 343235 seconds, and 343236 seconds of GPS week 1945, respectively. This z-curve is sweeping about 500 meters within 1 second, across the downtown of San Francisco, U.S.A. For validation, we take z-curve points in TOW 343235 seconds and migrate them into next second (marked as the white dots), using the three steps in Section V.5. The white dots exactly overlap on the black line, which proves the correctness of the proposed migration functions.

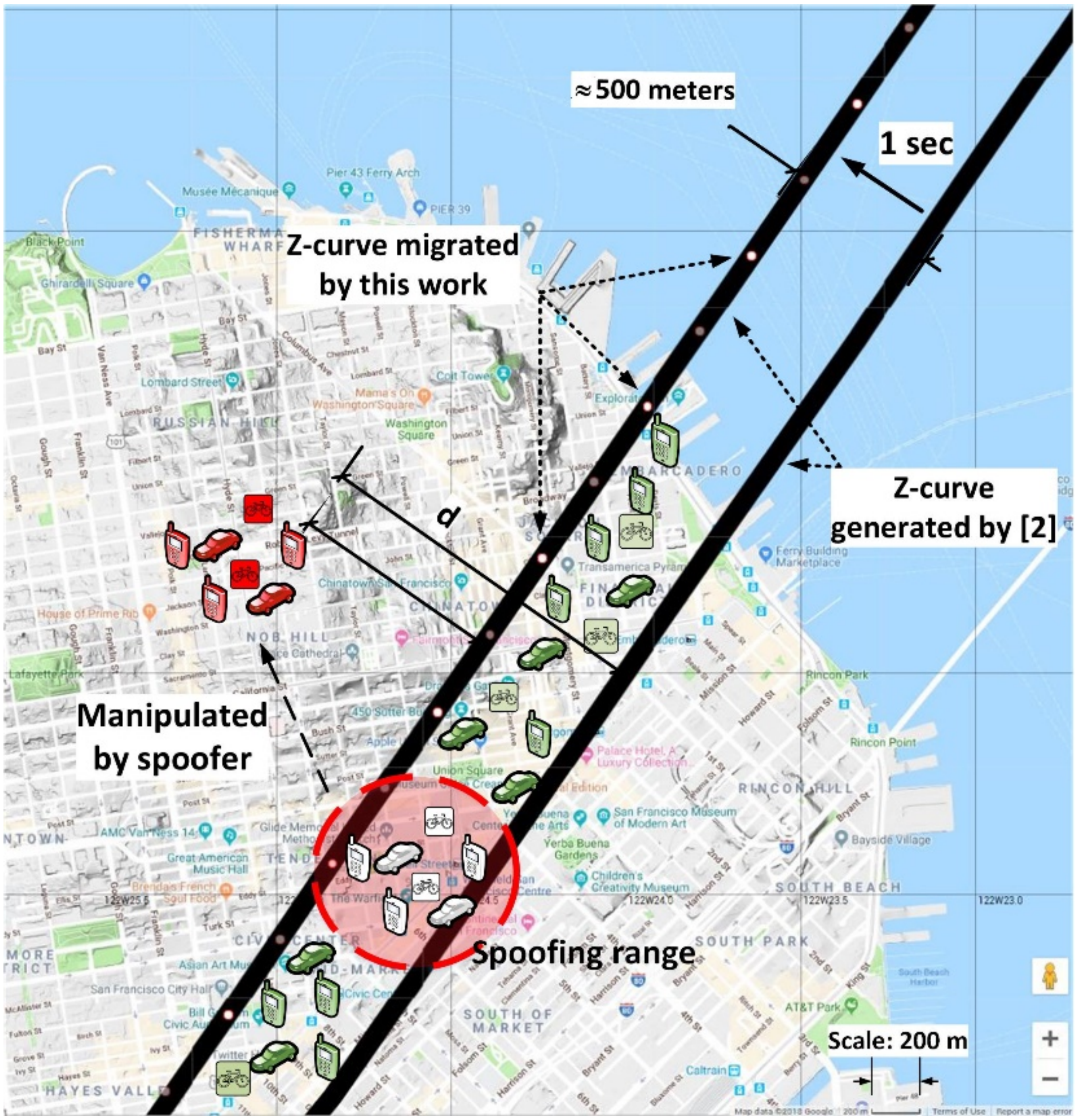


Figure V-8 Validation of z-curve migration function

Using Figure V-8, I illustrate an unverified potential application. Location-aware devices (smart phones, vehicles, bicycles, etc.) should sense and report the crossing of this z-curve (i.e. a zero crossing), as expected. Specifically, they would measure $\rho_6 \leq \rho_{15}$ at TOW 343235 seconds and $\rho_6 > \rho_{15}$ at next second. Normal devices are expected so as those in green. At the same time, abnormal devices may report the zero crossing at the red locations, while their physical locations are within these two z-curves. This or similar cases happen when a GNSS spoofer broadcasts a signal mimicking GPS satellites and inject errors into pseudoranges to mislead GNSS devices nearby. This scenario is partially verified in Section III.6 where timing errors in pseudoranges cause CORS stations measuring the zero crossing at wrong positions. To fully realize this application, more users' data are required. The essential reason for the effectiveness of using z-curve on anti-spoofing is that spoofing signal has to spread over many receivers in a region and the zero crossing is a deterministic event independent of any GNSS receivers. More crowd source spatial-temporal applications are expected to develop upon these z-curves, thanks to its deterministic and universal properties.

Visualization

This section aims to show more results of z-curve migration, to illustrate the process and a drawback. In Figure V-9, I plot the oval coverages of two satellites (PRN 2 and PRN 8) generated by the TSV model in Chapter IV, their z-curve at TOW 347500, and their z-curve at 30 minutes ago. Clearly, the z-curve grows longer, because the overlapping coverage area increases during this period. From a short z-curve to its longer version, our algorithm inherently transforms the curve points to account for the pose of z-surface, aligns each point to the Earth surface, and finally extends in both side until reaching the boundary of coverage. This process enables connections between curve points at old and new timestamps, which are represented as black lines between two z-curves.

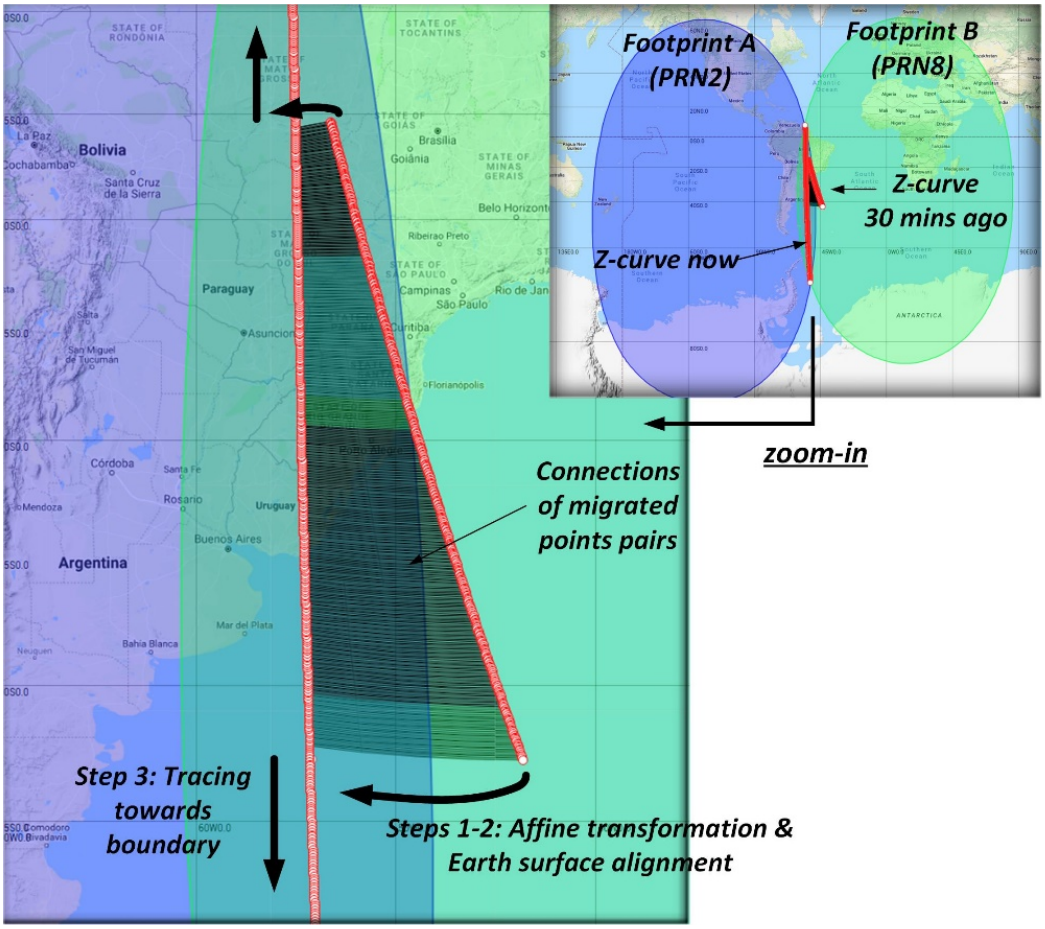

Footprint A (PRN2)
Footprint B (PRN8)
Z-curve 30 mins ago
Z-curve now
zoom-in
Connections of migrated points pairs
Step 3: Tracing towards boundary
Steps 1-2: Affine transformation & Earth surface alignment
Bolivia
Paraguay
Argentina
Uruguay

Figure V-9 Migration of a z-curve (PRN 2, PRN 8) from TOW 345700 to 347500 (30 minutes) at GPS week 1945

In Figure V-10, the z-curve (PRN 6 and PRN 15) is shrinking in length during this 30-minute period, because two satellites are flying away from each other and the co-coverage area becomes smaller. In this case, my algorithm has to trim out curve points that are out of coverage boundary, and does not need to run the PISI solver for curve extension. In Figure V-11, I show the validity of our migration function in a 2-hour period. Starting from “now” (TOW 343000), we migrate the curve to states at 15 minutes, and 30 minutes ago, as well as to states in next 90 minutes. The proposed algorithm describes the dynamic behavior in a very smooth way.

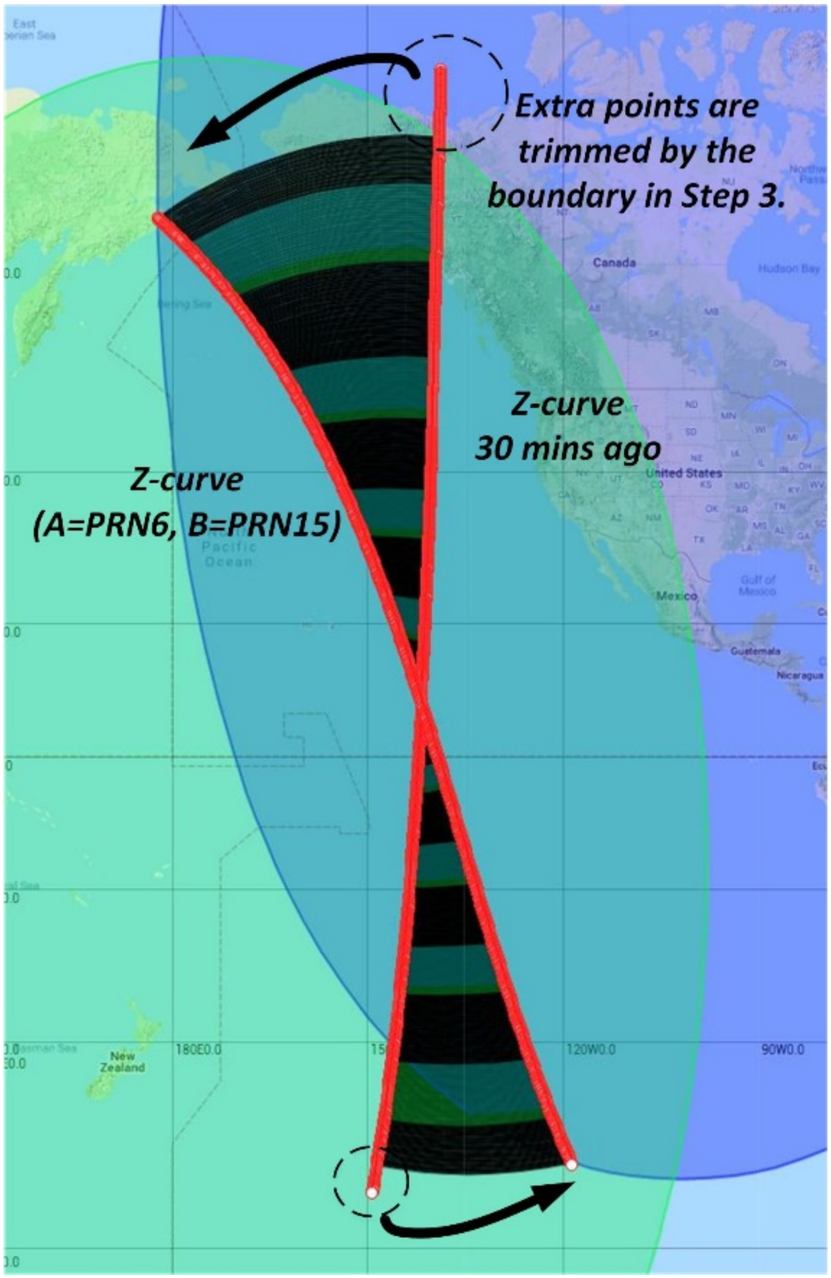


Figure V-10 Migration of a z-curve (PRN 6, PRN 15) from TOW 345700 to 347500 (30 minutes) at GPS week 1945

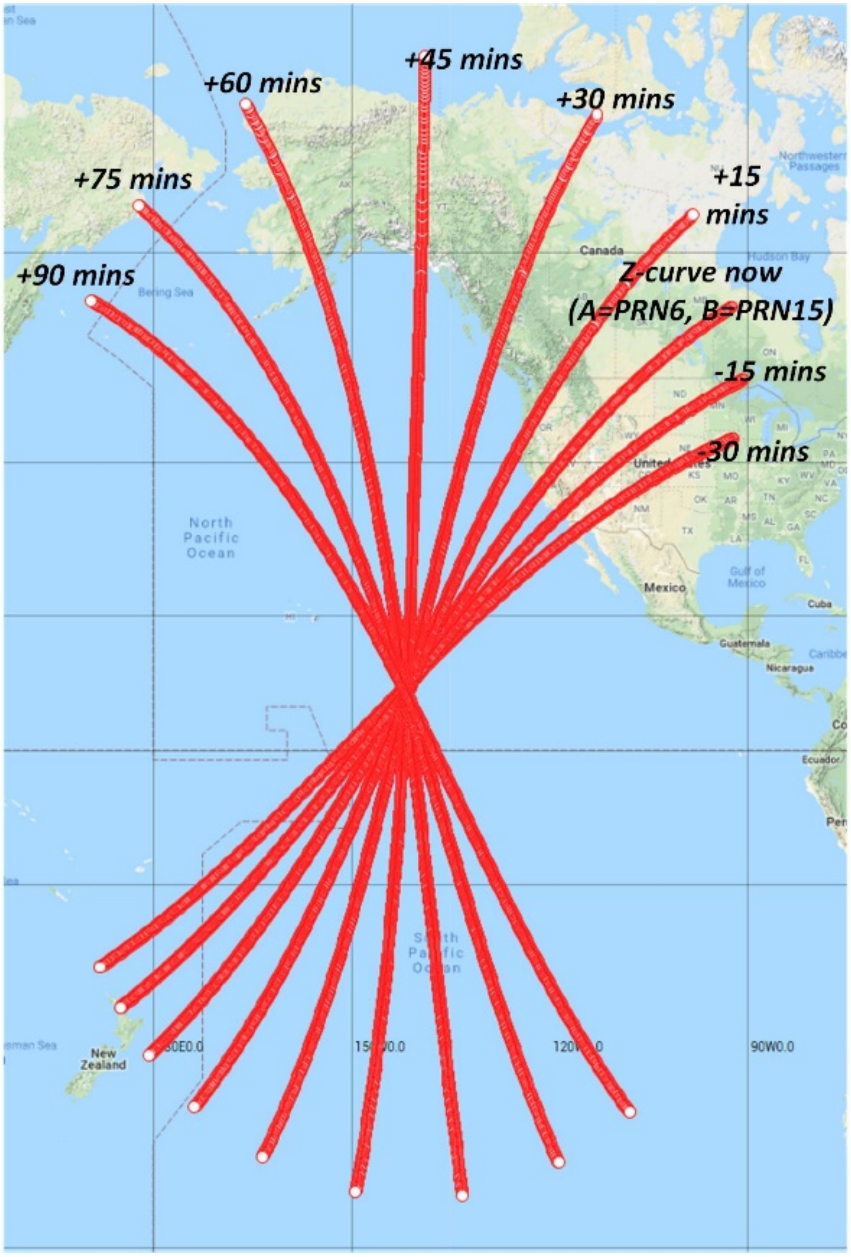


Figure V-11 Migrations of a z-curve (PRN 6, PRN 15) at every 15 minutes during 2 hours

In Figure V-12, a drawback of this proposed migration method is illustrated. The z-curve (PRN 8 and PRN 23) rotates clockwise. At half hour ago, one end of the z-curve is sparser than its other portions. This occurs because the curve points are generated using solver in Section III.5

whose tracing parameter is the latitude. Therefore, less curve points will be generated when the z-curve is more parallel to a latitude line. This sparsity will be maintained in the subsequent states, using the proposed migration function. The sparsity is even worsened in this particular case, where the z-curve becomes nearly perpendicular to latitude lines. The densities of migrated points and new extended points are not even in this case. Due to page limit, I will keep mitigation of this drawback in our future works.

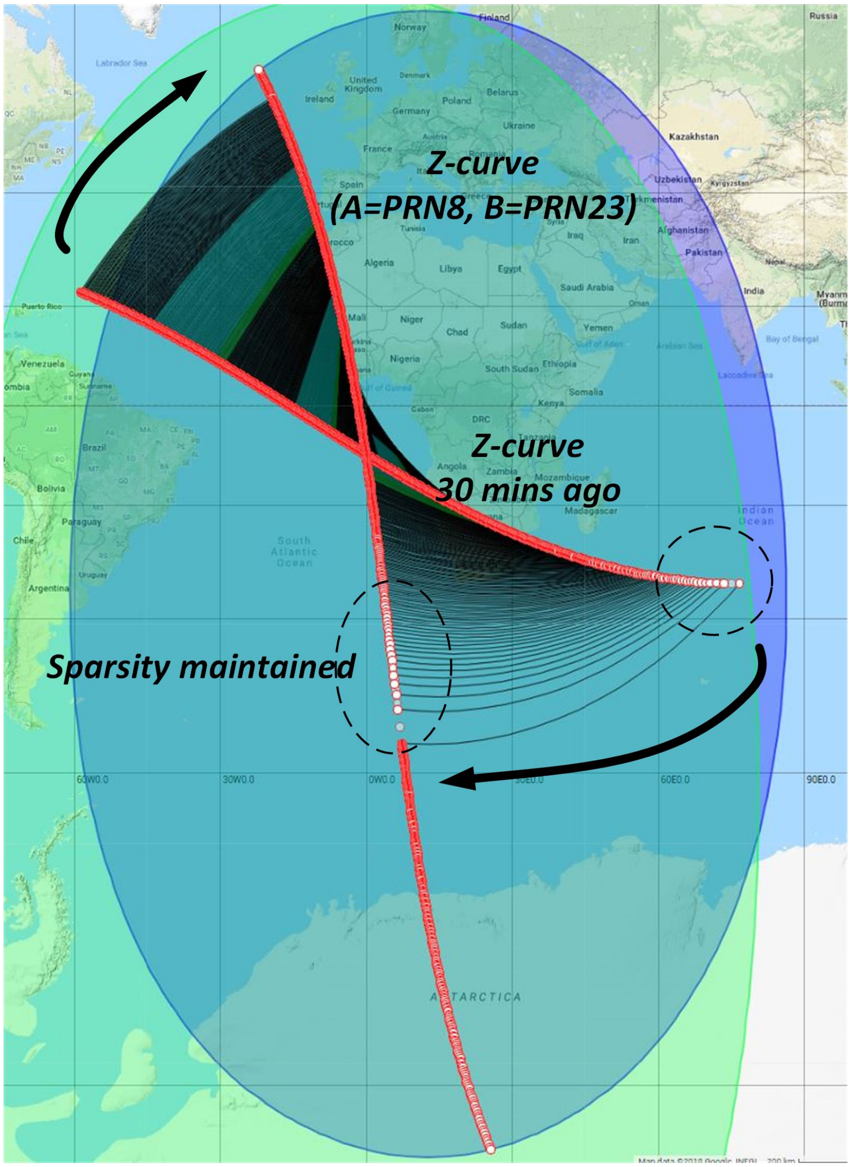


Figure V-12 Migration of a z-curve (PRN 8, PRN 23) from TOW 345700 to 347500 (30 minutes) at GPS week 1945

V.7 Conclusion

This chapter models the migrations of the satellite TSV, *z(k)*-surface, and *z(k)*-curve. Their physical movements are precisely described as functions of time and satellite position. Z-surface migration is composed of translation, rotation, scaling, and reflection, as a result of the co-movement of two satellites. The migration of z-curve is solved numerically by transforming along z-surface migration, aligning to Earth surface via Newton’s method, and trimming/extending points to coverage boundary via a PISI solver. Experimental results show that the proposed migration models can produce migrated z-curve points to match the validated solution given in Chapter III exactly, which also implies the correctness of z-surface migration model. The speed of GPS z-curves is analyzed using the proposed model. The proposed geometric and numerical approaches may be also applicable to other implicit and parametric shape problems. The theory of z-curve may serve as a system of spatiotemporal reference grids for the development of more location and time based applications.

CHAPTER VI CONCLUSION

VII.1 Contributions

The main contribution of this dissertation is the development and validation of TDOA based z-surface projection theory of GNSS signal, time, and position. My efforts present here answer the following questions regarding z-surface/curve. Firstly, this work provides mathematical tools to analyze a receiver's relationship with a z-surface, in terms of geometry and measurement. Secondly, this work formalizes the Cartesian model and properties of z-surface, as well as its extension to k-surface. Thirdly, this work provides parameterization algorithms for z/k-curve at the reference ellipsoid. Finally, this work analyzes the motion behavior of z/k-surface and derives the migration functions of z-surface and z-curve. All these components have been validated with measurement and orbital data collected by thousands of survey-grade reference stations in CORS and IGS networks.

The second contribution of this dissertation is a precise and efficient tool for satellite coverage analysis. Not only GNSS satellites but also other LEO, MEO, and GEO satellites, are compatible with the proposed models of Terrestrial Service Volume (or even Space Service Volume). This work also presents an algebraic parameterization algorithm to obtain the boundary of coverage footprint. The validated results prove that the previous popular assumption of sphere Earth leads to noticeable area errors of signal coverage analysis.

The third contribution of this dissertation is the development of an efficient software solution for GNSS urban positioning. Its geometric model of urban environment (buildings, trees) is proved efficient and effective. This work uses the R-Tree as a spatial partitioning data structure to organize urban elements, which saves the computation time and storage for satellite selection. The proposed software solution is validated and practical.

VII.2 Implications and Future Works

As digital devices become more and more intelligent, the trend of tightly coupling between GNSS and computing science will be realized sooner. I believe this dissertation is just a starting point of geometric models in GNSS signals, for next generation of computational GNSS. This work builds the interface using geometric tools from GNSS signals and receivers to computation models. It analyzes the inside principles of the physical layer and the measurement layer of GNSS, which then become base for geometric models of the computational layer of GNSS. The proposed z-curve theory may serve as a new foundation for tightly coupling between GNSS and computation, because it has three major new properties: a common event that can be measured by a group of GNSS users, a deterministic model that contains precise GNSS information, and a low-latency observation that every GNSS receiver can sense it efficiently. Hence, advanced applications such as integrity monitoring, geodetic and atmospheric surveys may be developed upon it.

Future works can be outlined as follows. First, the entire z-curve theory could be extended to carrier phase pseudoranges. This work only analyzes code pseudorange, which has lower resolution yet more popularity than carrier pseudorange. With carrier pseudorange enabled, the z-curve may enable observations of atmospheric delays, multipath delays, etc. Second, a crowd-source application scenario with z-curves should be developed, to demonstrate the practical power of the proposed models. For instance, one of the most important future works is to implement the scenario shown in Figure V-8, which makes the z-curve a reference grid for integrity monitoring that is globally available and highly deterministic. Third, other characteristics of z-curve/surface should be explored. For example, its thickness is related to the uncertainty of pseudoranges, which can be translated to the duration of zero crossing.

REFERENCES

[1] Lamport, Leslie. "Time, clocks, and the ordering of events in a distributed system." *Communications of the ACM* 21, no. 7 (1978): 558-565.

[2] Levin, Joshua. "A parametric algorithm for drawing pictures of solid objects composed of quadric surfaces." *Communications of the ACM* 19, no. 10 (1976): 555-563.

[3] Bajaj, Chandrajit L., Christoph M. Hoffmann, Robert E. Lynch, and J. E. H. Hopcroft. "Tracing surface intersections." *Computer aided geometric design*5, no. 4 (1988): 285-307.

[4] Miller, James R., and Ronald N. Goldman. "Geometric algorithms for detecting and calculating all conic sections in the intersection of any 2 natural quadric surfaces." *Graphical Models and Image Processing* 57, no. 1 (1995): 55-66.

[5] Wang, Wenping, Ronald Goldman, and Changhe Tu. "Enhancing Levin's method for computing quadric-surface intersections." *Computer Aided Geometric Design* 20, no. 7 (2003): 401-422.

[6] Dupont, Laurent, Sylvain Lazard, Sylvain Petitjean, and Daniel Lazard. "Towards the robust intersection of implicit quadrics." In *Uncertainty in Geometric Computations*, pp. 59-68. Springer US, 2002.

[7] Dupont, Laurent, Daniel Lazard, Sylvain Lazard, and Sylvain Petitjean. "Near-optimal parameterization of the intersection of quadrics." In *Proceedings of the nineteenth annual symposium on Computational geometry*, pp. 246-255. ACM, 2003.

[8] Yefimov, Nikolai Vladimirovich. *Quadratic forms and matrices: an introductory approach*. Academic Press, 2014.

[9] Hoschek, Josef, Dieter Lasser, and Larry L. Schumaker. *Fundamentals of computer aided geometric design*. AK Peters, Ltd., 1993.

[10] Heo, Hee-Seok, Myung-Soo Kim, and Gershon Elber. "The intersection of two ruled surfaces." *Computer-Aided Design* 31, no. 1 (1999): 33-50.

[11] Puig-Pey, Jaime, Akemi Gálvez, and Andrés Iglesias. "A new differential approach for parametric-implicit surface intersection." In *International Conference on Computational Science*, pp. 897-906. Springer Berlin Heidelberg, 2003.

[12] Mehlhorn, Kurt and C. Yap. (Dec. 2006). “Lecture 21—Surfaces.” [Online]. Available: http://cs.nyu.edu/yap/bks/egc/09/21Surfaces.pdfs

[13] Bucher, Ralph, and D. Misra. "A synthesizable VHDL model of the exact solution for three-dimensional hyperbolic positioning system." Vlsi Design15, no. 2 (2002): 507-520.

[14] Fang, Bertrand T. "Simple solutions for hyperbolic and related position fixes." *IEEE transactions on aerospace and electronic systems* 26, no. 5 (1990): 748-753.

[15] Bensky, Alan. *Wireless positioning technologies and applications*. Artech House, 2016.

[16] Sirola, Niilo. "Closed-form algorithms in mobile positioning: Myths and misconceptions." In *Positioning navigation and communication (WPNC), 2010 7th Workshop on*, pp. 38-44. Ieee, 2010.

[17] Bhatti, Jahshan A., Todd E. Humphreys, and Brent M. Ledvina. "Development and demonstration of a TDOA-based GNSS interference signal localization system." In *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pp. 455-469. IEEE, 2012.

[18] Wikipedia contributors. "Reference ellipsoid." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 26 Mar. 2015. Web. 24 Jun. 2015.

[19] Maling, Derek Hylton. *Coordinate systems and map projections*. Elsevier, 2013.

[20] Wikipedia contributors, "Digital elevation model," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=Digital_elevation_model&oldid=715977302 (acc essed June 2, 2016).

[21] Seeber, Günter. *Satellite geodesy: foundations, methods, and applications*. Walter de Gruyter, 2003.

[22] Agnew, Duncan Carr, and Kristine M. Larson. "Finding the repeat times of the GPS constellation." *GPS solutions* 11, no. 1 (2007): 71-76.

[23] National Geodetic Survey. “Continuously Operating Reference Station (CORS)”. Retrieved from http://geodesy.noaa.gov/CORS/, Mar 27, 2016.

[24] Fu, Guoyu, Liu, Jyh-Charn, "Order of Pseudoranges (OOP) of GNSS: Spatial Modeling and Analysis," *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, Tampa, Florida, September 2015, pp. 3370-3382.

[25] Kaplan, Elliott, and Christopher Hegarty, eds. “*Understanding GPS: principles and applications*.” Artech house, 2005.

[26] Misra, Pratap, and Per Enge. *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press, 2006.

[27] Interface Control Working Group. *IS-GPS-200 Revision H: Navstar GPS Space Segment/Navigation User Interfaces*. Apr 13, 2014. from: http://www. gps. gov/technical/icwg/IS-GPS-200H, pdf.

[28] Parkinson, B.W. et al. *GPS: theory and applications*, vols 1 and 2. American Institute of Aeronautics and Astronautics, Washington, DC, 1996.

[29] Marquis, Willard A., and Daniel L. Reigh. "The GPS Block IIR and IIR-M Broadcast L-band Antenna Panel: Its Pattern and Performance." *Navigation*62, no. 4 (2015): 329-347.

[30] Bauer, F. H., Michael C. Moreau, M. E. Dahle-Melsaether, W. P. Petrofski, B. J. Stanton, S. Thomason, G. Al Harris, R. P. Sena, and L. Parker Temple III. "The GPS space service volume." (2006).

[31] Federal Aviation Administration. “Satellite Navigation - GPS - Policy - Selective Availability.” *FAA Seal*, Nov. 2014. Available at: www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/policy/availability/.

[32] Cohenour, Curtis, and FRANK GRAAS. "GPS orbit and clock error distributions." *Navigation* 58, no. 1 (2011): 17-28.

[33] Warren, David LM, and John F. Raquet. "Broadcast vs. precise GPS ephemerides: a historical perspective." *GPS Solutions* 7, no. 3 (2003): 151-156.

[34] Montenbruck, Oliver, Peter Steigenberger, and André Hauschild. "Broadcast versus precise ephemerides: a multi-GNSS perspective." *GPS Solutions* 19, no. 2 (2015): 321-333.

[35] Salós, Daniel, Christophe Macabiau, Anaïs Martineau, Bernard Bonhoure, and Damien Kubrak. "Nominal GNSS pseudorange measurement model for vehicular urban applications." In *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pp. 806-815. IEEE, 2010.

[36] Heng, Liang, Grace Xingxin Gao, Todd Walter, and Per Enge. "Statistical characterization of GPS signal-in-space errors." In *Proceedings of the 2011 International Technical Meeting of the Institute of Navigation (ION ITM 2011), San Diego, CA*, pp. 312-319. 2011.

[37] Klobuchar, John A. "Ionospheric time-delay algorithm for single-frequency GPS users." *Aerospace and Electronic Systems, IEEE Transactions on* 3 (1987): 325-331.

[38] Hopfield, H. S. "Two-quartic tropospheric refractivity profile for correcting satellite data." *Journal of Geophysical research* 74.18, 1969: 4487-4499.

[39] Saastamoinen, J. "Contributions to the theory of atmospheric refraction." *Bulletin Géodésique* (1946-1975) 107.1, 1973: 13-34.

[40] Warner, Jon S. and R. Johnston, "A Simple Demonstration that the Global Positioning System (GPS) is Vulnerable to Spoofing," *Journal of Security Administration*, vol. 25, pp. 19–28, 2002.

[41] Humphreys, Todd E., Brent M. Ledvina, Mark L. Psiaki, Brady W. O'Hanlon, and Paul M. Kintner Jr. "Assessing the spoofing threat: Development of a portable GPS civilian spoofer." In *Proceedings of the ION GNSS international technical meeting of the satellite division*, vol. 55, p. 56. 2008.

[42] Lian, Zhenyu, Y. Tan, Y. Xu, J. Li. "Static and dynamic models of observation toward earth by agile satellite coverage." In *Proceedings of International Workshop on Planning and Scheduling for Space*. Darmstadt, Germany: ESOC 2011, pp. 1-6.

[43] Zhang, Pengfei, C. Xu, C. Hu, Y. Chen. "Research on Coverage Performance of GNSS." *Przegląd Elektrotechniczny*, *89*(1b), 2013, pp.67-70.

[44] Chiyangcabut, Sarawoot. "An Analysis of a Satellite Surveillance System." NAVAL POSTGRADUATE SCHOOL MONTEREY CA; 1999 March.

[45] Beste, D.C. "Design of satellite constellations for optimal continuous coverage." *IEEE Transactions on Aerospace and Electronic Systems*, (3), 1978, pp.466-473.

[46] Ulybyshev, Yuri. "Geometric analysis and design method for discontinuous coverage satellite constellations." *Journal of Guidance, Control, and Dynamics*, *37*(2), 2014, pp.549-557.

[47] Wilkinson, C. K. "Coverage regions: how they are computed and used." *Journal of the Astronautical Sciences*, *42*, 1994, pp.47-70.

[48] Washburn, A.R. "Earth Coverage by Satellites in Circular Orbit." Department of Operations Research, NAVAL POSTGRADUATE SCHOOL, 2004.

[49] Chobotov, V.A. "Orbital Coverage." *Orbital mechanics*. American Institute of Aeronautics and Astronautics, 2002, pp.411-452.

[50] Cakaj, B.S., Kamo, A. Lala, and A. Rakipi. "The coverage analysis for low earth orbiting satellites at low elevation." *International Journal of Advanced Computer Science and Applications (IJACSA)*, *5*(6), 2014.

[51] Siocos, C.A. "Broadcasting-Satellite Coverage-Geometrical Considerations." *IEEE Transactions on Broadcasting*, (4), pp.84-87, 1973.

[52] Ott, R.H. "Vector analysis and satellite footprints." *IEEE Transactions on Geoscience and Remote Sensing*, *28*(1), 1990, pp.117-122.

[53] Weisstein, E. W. "Ellipsoid." *From MathWorld—A Wolfram Web Resource.* Accessed: Jul. 17, 2018. [Online]. Available: http://mathworld.wolfram.com/Ellipsoid.html

[54] Foley, J. D., A. Van Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips. "Introduction to computer graphics." Addison-Wesley. 1994.

[55] Weisstein, E. W. "Plane." *From MathWorld--A Wolfram Web Resource.* Accessed: Jul. 17, 2018. [Online]. Available: http://mathworld.wolfram.com/Plane.html

[56] Information and Analysis Center for Positioning, Navigation and Timing. "Orbital Constellation", Beidou Global Navigation Satellite System, Korolyov, Russia, https://www.glonass-iac.ru/en/guide/beidou.php (accessed October 8, 2018).

[57] Iriarte, J. C., I. Ederra, R. Gonzalo, Y. Brand, A. Fourmault, et al. "EBG superstrate array configuration for the WAAS space segment." *IEEE Transactions on Antennas and Propagation*, *57*(1), 2009, pp.81-93.

[58] Orbcomm LLC. *ORBCOMM System Overview, A80TD0008 – Revision G*, December 18, 2001.

[59] Kramer, H. J. "Iridium NEXT - Satellite Missions - eoPortal Directory." *Available at:* https://earth.esa.int/web/eoportal/satellite-missions/i/iridium-next [Accessed 8 Oct. 2018].

[60] Geoborders Satellite Ltd. (2018). "*IRIDIUM - How it works*." *Iridium.it. Available at*: http://www.iridium.it/en/iridium.htm [Accessed 8 Oct. 2018].

[61] N2yo.com. *Orbcomm satellites*. Available at: https://www.n2yo.com/satellites/?c=16&srt=5&dir=1 [Accessed 8 Oct. 2018].

[62] Wikipedia contributors, "List of GPS satellites," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=List_of_GPS_satellites&oldid=861901141 (accessed October 8, 2018).

[63] MATLAB and Mapping Toolbox Release 2018a, The MathWorks, Inc., Natick, Massachusetts, United States.

[64] Wikipedia contributors, "List of countries and dependencies by area," *Wikipedia, The Free Encyclopedia,* https://en.wikipedia.org/w/index.php?title=List_of_countries_and_dependencies_by_area&oldid=862203976 (accessed October 8, 2018).

[65] Celestrak.com. "*CelesTrak: Current NORAD Two-Line Element Sets*." Available at: https://www.celestrak.com/NORAD/elements/ (Accessed 9 Oct. 2018).

[66] Rhodes, B. *sgp4*.PyPI. Available at: https://pypi.org/project/sgp4/ (Accessed 9 Oct. 2018).

[67] Fu, Guoyu, C. Riedel, T. Holmes, and J. C. Liu. "Geometric Modeling of the Z-Surface and Z-Curve of GNSS Signals and Their Solution Techniques." *IEEE Transactions on Geoscience and Remote Sensing*, (99), pp.1-12, 2018.

[68] Murray, Richard M. *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[69] The-Crankshaft Publishing. "Geocentric Earth-Fixed Coordinate Systems (GPS)." *What-when-how: in depth tutorials and information*. Accessed: Sept. 26, 2018. [Online]. Available: http://what-when-how.com/gps/geocentric-earth-fixed-coordinate-systems-gps/.

[70] Burden, Richard L., and J. Douglas Faires. *Numerical analysis*. *Cengage Learning* 9, 2010.

[71] Mortenson, Michael E. *Geometric modeling*. 3rd ed. *New York: Industrial Press*, 2006.

[72] Wikipedia contributors. Great-circle distance. *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 14 Sep. 2018. Web. 27 Sep. 2018.

[73] Homeier, Herbert H. H. "A modified Newton method with cubic convergence: the multivariate case." *Journal of Computational and Applied Mathematics* 169, no. 1 (2004): 161-169.

[74] Weisstein, Eric W. Newton's Method. From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/NewtonsMethod.html

[75] Dehghanian, Vahid, J. Nielsen, and G. Lachapelle, "GNSS spoofing detection based on signal power measurements: statistical analysis." *International Journal of Navigation and Observation*, vol. 18, 2012.

[76] Corbett, James C., Jeffrey Dean, Michael Epstein, Andrew Fikes, Christopher Frost, J. J. Furman, Sanjay Ghemawat et al. "Spanner: Google's globally distributed database." *ACM Transactions on Computer Systems* (TOCS) 31, no. 3 (2013): 8.

[77] Esfahani, Mehrnaz S. Security analysis of phasor measurement units in smart grid communication infrastructures, Master's thesis, University of Nebraska - Lincoln, Lincoln, Nebraska, 5 2014.

[78] Psiaki, Mark L., Brent W. O'Hanlon, Jahshan A. Bhatti, Daniel P. Shepard, and Todd E. Humphreys. "GPS spoofing detection via dual-receiver correlation of military signals." *IEEE Transactions on Aerospace and Electronic Systems* 49, no. 4 (2013): 2250-2267.

[79] Heng, Liang, Daniel Chou, and Grace Xingxin Gao. "Cooperative GPS signal authentication from unreliable peers." In *27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2014), Tampa, FL,* USA, 2801-2809. 2004.

[80] Heng, Liang, Jonathan J. Makela, Alejandro D. Dominguez-Garcia, Rakesh B. Bobba, William H. Sanders, and Grace Xingxin Gao. "Reliable GPS-based timing for power systems: A multi-layered multi-receiver architecture." In *Power and Energy Conference at Illinois (PECI), 2014*, 1-7. IEEE, 2014.

[81] Google LLC. *Wi-Fi location: ranging with RTT | Android Developers*. [online] Android Developers. Available at: https://developer.android.com/guide/topics/connectivity/wifi-rtt [Accessed 17 Oct. 2018].

[82] Bensky, Alan. *Wireless Positioning Technologies and Applications*. Norwood, MA, USA: Artech House, 2016.

[83] Hilbert, David and Stefan Cohn-Vossen, *Geometry and the Imagination*, no. 87. Providence, RI, USA: AMS, 1999.

[84] E. Farin, Gerald. *Curves and Surfaces for CAGD: A Practical Guide*. 5-th edition, Academic Press, 2002.

[85] E. Farin, Gerald, J. Hoschek, and M. S. Kim, eds. *Handbook of computer aided geometric design*. Elsevier, 2002.

[86] International GNSS Service. IGS Network. Accessed: Mar. 27, 2016. [Online]. Available: http://www.igs.org/network

[87] CDDIS (Crustal Dynamics Data Information System) (2003) FTP server ftp://cddisa.gsfc.nasa.gov/pub/gps/gpsdata/brdc/

[88] OpenGL. Open Graphics Library. Accessed: Jul. 17, 2018. [Online]. Available: http://www.opengl.org

[89] CGAL. Computational Geometry Algorithms Library. Accessed: Jul. 17, 2018. [Online]. Available: http://www.cgal.org

[90] Lee, Young C. "Analysis of range and position comparison methods as a means to provide GPS integrity in the user receiver." In *Proceedings of the 42nd Annual Meeting of the Institute of Navigation*, pp. 1-4. 1986.

[91] Swokowski, E.W. *Calculus with analytic geometry*. Taylor & Francis, 1979.

[92] Wooton, William, E. Beckenbach, and F. Fleming. *Modern Analytic Geometry*. Houghton Mifflin Harcourt (HMH), 1983).

[93] Ferguson, C. “Intersections of ellipsoids and planes of arbitrary orientation and position.” *Journal of the International Association for Mathematical Geology* 11.3, 1979: 329-336.

[94] Klein, P. “On the Intersection Equation of A hyperboloid and A Plane.” *Applied Mathematics*, 2013, 4: 40-49.

[95] Gurtner, W. and L. Estey. *RINEX-The Receiver Independent Exchange Format-Version 3.00*. Astronomical Institute, University of Bern and UNAVCO, Boulder, Colorado, 2007.

[96] Spofford, P. and B. Remondi. *The national geodetic survey standard GPS format SP3* (SP3-a format). Available from the IGS website: http://igscb.jpl.nasa.gov/igscb/data/format/sp3_docu.txt (1994).

[97] Wikipedia Contributors. “Latitude.” Wikipedia, the Free Encyclopedia. Accessed: Nov. 13, 2017. Available: https://en.wikipedia.org/w/index.php?title=Latitude&oldid=809055522.

[98] Nature, International Journal of Science. “A SpaceX Rocket Blows a Hole in the Atmosphere.” Accessed: Mar. 12, 2018. [Online]. Available: https://www.nature.com/articles/d41586-018-02967-2

[99] European Global Navigation Satellite Systems Agency. “GNSS Market Report”. GSA, Issue 5, 2017. Available at: https://www.gsa.europa.eu/market/market-report.

[100] Van Diggelen, Frank. *A-GPS: Assisted GPS, GNSS, and SBAS.* Artech House, 2009.

[101] Wikipedia contributors, "GLONASS," Wikipedia, The Free Encyclopedia, https://en.wikipedia.org/w/index.php?title=GLONASS&oldid=876332426 (ac cessed January 9, 2019).

[102] U.S. National Coordination Office for Space-Based Positioning Navigation and Timing, Jul. 2017, [online] Available: https://www.gps.gov/policy/legislation/loran-c/.

APPENDIX 1

Substituting the Parametric Earth (III.13) into $x^T M_Z x + N_Z x + L_Z = 0$ in (III.11)

I first expand $\mathbf{x}^T M_Z \mathbf{x} + N_Z \mathbf{x} + L_Z = 0$ as:

$$m_{1,1}x^2 + m_{2,2}y^2 + m_{3,3}z^2 + 2m_{1,2}xy + 2m_{1,3}xz + 2m_{2,3}yz + n_1x + n_2y + n_3z + l = 0. \quad \text{(A1.1)}$$

When (III.13) is substituted into (A1.1), we obtain constraints $W(u)$ and $E(u)$ for west and east hemispheres, respectively:

$$\begin{cases} W(u) = \alpha_1 u^2 + \alpha_2 u - (\alpha_3 u + \alpha_4)\sqrt{1-u^2} + \alpha_5 = 0, & \lambda \in [-\pi, 0), \\ E(u) = \alpha_1 u^2 + \alpha_2 u + (\alpha_3 u + \alpha_4)\sqrt{1-u^2} + \alpha_5 = 0, & \lambda \in [0, \pi). \end{cases} \quad \text{(A1.2)}$$

where $u = \cos(\lambda)$, and coefficients are given as functions of the tracing parameter φ:

$$\alpha_1 = a_E^2 cos^2(\varphi)\big(m_{1,1} - m_{2,2}\big),$$

$$\alpha_2 = a_E b_E m_{1,3}\sin(2\varphi) + a_E n_1 \cos(\varphi),$$

$$\alpha_3 = 2a_E^2 m_{1,2} cos^2(\varphi),$$

$$\alpha_4 = a_E b_E m_{2,3}\sin(2\varphi) + a_E n_2 \cos(\varphi),$$

$$\alpha_5 = a_E^2 m_{2,2} cos^2(\varphi) + b_E^2 m_{3,3} sin^2(\varphi) + b_E n_3 \sin(\varphi) + l.$$

Eliminating the square root operator in both equations in (A1.2) will give us the following equation:

$$F(u):\ \beta_1 u^4 + \beta_2 u^3 + \beta_3 u^2 + \beta_4 u + \beta_5 = 0, \quad \text{(A1.3)}$$

where $u \in [-1,1]$, and the coefficient β's are listed as below:

$$\beta_1 = \alpha_1^2 + \alpha_3^2, \quad \beta_2 = 2\alpha_3\alpha_4 + 2\alpha_1\alpha_2,$$

$$\beta_3 = \alpha_2^2 - \alpha_3^2 + \alpha_4^2 + 2\alpha_1\alpha_5,$$

$$\beta_4 = 2\alpha_2\alpha_5 - 2\alpha_3\alpha_4, \quad \beta_5 = \alpha_5^2 - \alpha_4^2.$$

Roots of $W(u) = 0$ (and $E(u) = 0$) are a subset of roots of $F(u) = 0$, whereas some of roots of $F(u) = 0$ may not satisfy $E(u) = 0$ or $W(u) = 0$ at the same time. As such, we first find real roots of $F(u) = 0$ and test them one by one in W and E. The algebraic solutions for a 4th order polynomial equation are available long before, which takes up to a dozen scalar operations of square, cube, squared root, and cubic root. For the case of $\beta_1 = 0$, (A1.3) is degenerated to 3rd order.

APPENDIX 2

Solving Processing of $\alpha\cos(\lambda) + \beta\sin(\lambda) + \gamma = 0$

Step 1: If $\alpha = 0$ and $\beta = 0$ and $\gamma \neq 0$, there is no satisfying solution.

Step 2: Let $u = \cos(\lambda) \in [-1,1]$, and thus

$$\sin(\lambda) = \begin{cases} \sqrt{1-u^2}, & if\ \lambda \in [0,\pi), \\ -\sqrt{1-u^2}, & if\ \ \lambda \in [-\pi,0). \end{cases}$$

And the original equation can be rewritten to be:

$$\begin{cases} W(u) = \alpha u - \beta\sqrt{1-u^2} + \gamma = 0, & \lambda \in [-\pi,0), \\ E(u) = \alpha u + \beta\sqrt{1-u^2} + \gamma = 0, & \lambda \in [0,\pi). \end{cases}$$

W and E stand for west and east hemispheres, respectively, as defined by their corresponding longitude λ.

Step 3: Solve for potential solutions for $W(u) = 0$ and $E(u) = 0$, by solving their unified equation:

$$(\alpha^2 + \beta^2)u^2 + 2\alpha\gamma u + \gamma^2 - \beta^2 = 0.$$

The valid solution should be real numbers, and within $[-1,1]$.

Step 4: For each root u, use $\cos^{-1}(u)$ to solve for an appropriate (in west or east hemisphere) longitude λ. If $\frac{-\lambda-\alpha u}{\beta} > 0$, then the resulting λ should be within $[0,\pi)$. If $\frac{-\lambda-\alpha u}{\beta} < 0$, the resulting λ should be within $[-\pi,0)$. If $\frac{-\lambda-\alpha u}{\beta} = 0$, then there will be two λ's of identical magnitude, but one in west and the other in east hemisphere.