

---

Max Wendler

# Impact of Satellite Mobility Models on Vehicle-to-Satellite Communication

Master's Thesis in Computer Science

6 March 2024

Please cite as:

Max Wendler, "Impact of Satellite Mobility Models on Vehicle-to-Satellite Communication," Master's Thesis (Masterarbeit),  
Faculty of Computer Science, TU Dresden, Germany, March 2024.

# **Impact of Satellite Mobility Models on Vehicle-to-Satellite Communication**

Master's Thesis in Computer Science

vorgelegt von

**Max Wendler**

geb. am 9. April 1999  
in Großenhain

angefertigt an der

**Technischen Universität Dresden**  
**Fakultät Informatik**  
**Chair of Networked Systems Modeling**

Betreuer: **Mario Franke**  
Gutachter: **Christoph Sommer**  
**Burkhard Hensel**

Abgabe der Arbeit: **6. März 2024**

## Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.

Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

## Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.

I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

(Max Wendler)

Dresden, 6 March 2024

---

# Abstract

---

Due to short propagation delays of a few milliseconds and relatively low costs, the application of Low-Earth Orbit (LEO) satellites in vehicular networks is very attractive, e.g. for road safety and route optimization. Because the integration of air and ground traffic with very large LEO constellations, which are needed for continuous and global coverage, results in highly complex, dynamic network topologies, evaluation of such networks requires comprehensive simulation that models movement in space, air and on the ground in detail. The *OMNeT++* - based open source simulation framework *space\_Veins* achieves this, but can currently only model existing satellites using the relatively accurate Simplified General Perturbations 4 (SGP4) model. For simulating planned constellations, alternative orbit models are required. This thesis implements two simplified models enabling this: One based on Kepler's two-body problem, and one that simplifies it further by approximating all orbits as circular. It was evaluated if the models are valid alternatives to SGP4 in *space\_Veins* despite their simplifications, by assessing how they differ from it in satellite positions, propagation delays and satellite availability. While the circular model can even produce more accurate positions, and delay differences are similar for both models, the evaluation of availability proves the circular model invalid as a general alternative. The Keplerian model generally produces relatively small and expectable deviations for communication, and thus is deemed a valid alternative. However ultimately, the significance of its deviations and validity depend on the applications it would be used for.

---

## Kurzfassung

---

Da Satelliten in Low-Earth Orbits (LEOs) Funkübertragungszeiten von wenigen Millisekunden bieten und relativ günstig sind, sind sie attraktive Komponenten für Fahrzeug-Netzwerke, z.B. für Verkehrssicherheit und Routenoptimierung. Weil sich aus der Integration von sehr großen LEO-Konstellationen, die für kontinuierliche und globale Verfügbarkeit benötigt werden, mit Boden- und Luftverkehr hochkomplexe, dynamische Netztopologien ergeben, erfordert die Evaluation solcher Netzwerke detaillierte Simulation der Mobilität am Boden, in der Luft und im All. Das OMNeT++-basierte Simulationsframework `space_Veins` realisiert dies, kann auf der Grundlage von dem relativ genauen Modell Simplified General Perturbations 4 (SGP4) momentan aber nur bereits existierende Satelliten modellieren. Um künftige Konstellationen zu simulieren, werden alternative Modelle benötigt. Diese Arbeit implementiert zwei solche, vereinfachte Modelle: Eines basierend auf Keplers Zweikörperproblem, und eines das es weiter vereinfacht, indem alle Orbits als Kreise modelliert werden. Die Modelle wurden darauf untersucht, ob sie trotz ihrer Vereinfachungen valide Alternativen zu SGP4 für `space_Veins` sind. Dafür wurde evaluiert, wie sich ihre Satellitenpositionen und die daraus folgenden Übertragungszeiten und Kommunikationsverfügbarkeiten von SGP4 unterscheiden. Während das Kreismodell mitunter sogar genauere Positionen berechnen kann, und die Unterschiede der Übertragungszeiten beider Modelle zu SGP4 ähnlich sind, wurde seine Validität durch die Überprüfung der Verfügbarkeit abgelehnt. Das Kepler'sche Modell führt dagegen zu relativ geringen und erwartbaren Änderungen der Kommunikation, so dass es als valide Alternative betrachtet wird. Schlussendlich hängt die Signifikanz seiner Auswirkungen und seine Validität jedoch von den konkreten Anwendungen ab, in denen es genutzt werden könnte.

---

# Contents

---

<b>Abstract</b>	<b>iii</b>
<b>Kurzfassung</b>	<b>iv</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Fundamentals</b>	<b>3</b>
2.1 space_Veins . . . . .	3
2.2 Keplerian orbit model . . . . .	7
2.3 Circular orbit model . . . . .	11
2.4 Spherical and polar coordinates . . . . .	12
<b>3 Related Work</b>	<b>15</b>
<b>4 Implementation</b>	<b>23</b>
4.1 Keplerian orbit model . . . . .	23
4.2 Circular orbit model . . . . .	26
<b>5 Evaluation</b>	<b>37</b>
5.1 Satellite constellations . . . . .	38
5.2 Surface locations . . . . .	45
5.3 General simulation parameters . . . . .	50
5.4 Simulation configuration via Snakemake . . . . .	50
5.5 Metrics . . . . .	52
5.6 Results . . . . .	55
<b>6 Conclusion</b>	<b>88</b>
<b>Bibliography</b>	<b>97</b>

---

## Chapter 1

# Introduction

---

Orbiting Earth at much smaller altitudes than satellites in other orbital regions, Low-Earth Orbit (LEO) satellites provide short propagation delays of a few milliseconds, enabling real-time communication applications, and can be launched at relatively low costs. Because of this, they have received enormous scientific and commercial interest recently [1]. In comparison to satellites on middle-earth orbits or geostationary orbits, each satellite can only cover a small surface area. Continuous and global coverage thus requires very large satellite constellations [2].

One area with much potential for LEO satellites are vehicular networks, where they can be used for e.g. road safety and route optimization [3]. More complex vehicular networks not only include communication with static and mobile nodes on the ground, but also aired vehicles and inter-satellite communication. Consequently, they are characterized by highly dynamic network topologies. Due to the complexity of very large Space-Air-Ground-Integrated Networks (SAGINs), accurate and efficient evaluation of their performance requires network simulators which comprehensively consider node movement in all three segments. The *OMNeT++* - based open source simulation framework *space\_Veins* [4] is one of the few works that model node movement in all segments *in detail*, and the only one to validate the accuracy of its approach of integrating them.

However, *space\_Veins* currently relies on the Simple General Perturbations 4 (SGP4) algorithm to model satellite orbits. SGP4 is a well-established solution for simulation via *orbit propagation*, but based on Two-Line Element set (TLE) data, which can only be created by observing existing orbits [5]. Hence, *space\_Veins* can currently only simulate existing satellites, not planned constellations. To enable this, alternative orbit models which can run from purely artificial inputs are required. This thesis aims to firstly, expand *space\_Veins* with two such models, which allow for rather intuitive specification of orbits; and secondly, to evaluate if they are valid alternatives to SGP4 for short simulation periods of 12 hours, by assessing

if they maintain reasonable accuracy for satellite positions and, more importantly, communication metrics. The models are: (1) A Keplerian orbit model based on Johannes Kepler's two-body problem [6], and (2) a circular orbit model that assumes perfectly circular orbits [7]. Compared to SGP4, both models are simplified and consider Earth's gravitation as the only reason for satellite movement on constant elliptical or circular orbits, instead of including e.g. radiation and gravitation of the Sun.

It has been shown that the Keplerian model thus predicts satellite positions with much higher positional error, which exceeds 1000 km after a few days [8]. However, evaluation of how significant these effects are in shorter simulation periods is scarce, and does not exist for communication parameters of individual ground-satellite links like availability and propagation delay. For example would a position error of 1000 km only cause a delay change of roughly 3 milliseconds at maximum, as radio waves travel close to the speed of light.

Assuming all orbits to be circular is an even bigger simplification, which has not been evaluated in detail yet. While it is to be expected that such an approximation of real elliptical orbits will perform worse than the Keplerian model, it might serve as a sufficient model of orbits with a nearly circular shape.

To validate the models' application in `space_Veins`, the major part of this thesis is a simulation study on position errors, satellite availability and propagation delay in comparison to SGP4. Also, runtime performance is evaluated. To compare the models' effects as representations of the same real satellites, methods are implemented to derive the Keplerian and circular parameters from a TLE. This also has the advantage of being completely based on the publicly available TLE data from NASA/NORAD [5], which includes important LEO constellations such as Starlink. From this data, five constellations with different orbital parameters are simulated over the period of 12 hours. As this thesis focuses on the differences caused by the satellite mobility models, it limits itself to simulating the satellites' communication w.r.t. to static ground stations, which are situated at three different latitudes. Hence, also the influence of Earth's oblate shape and thus non-uniform gravitation [9][10] is examined, which is part of SGP4 but not the other models.

This implementation of the Keplerian and circular model in `space_Veins` and their evaluation are presented in the following sections: Chapter 2 contains the fundamentals of `space_Veins`, the Keplerian and circular orbits, and of spherical and polar coordinates used for implementing the circular model. In Chapter 3, related work that employs orbit models for LEO satellite simulation and evaluates them is discussed. Chapter 4 presents the implementation of the orbit models in `space_Veins`. The simulation study to evaluate them, its metrics, results and their discussion are presented in Chapter 5. Chapter 6 finishes with a conclusion and an outlook on future work.



---

## Chapter 2

# Fundamentals

---

The fundamentals this thesis builds on are: (1) *space\_Veins*, which applies the SGP4 orbit model to simulate satellites from TLEs in a common coordinate system with ground and air vehicles. (2) The mathematical laws of the Keplerian and circular orbit models, and the parameters to specify concrete orbits in them. To be able to validate the models as alternatives to SGP4 by fully attributing changes in communication metrics to the change of the orbit model, the three models should use the same inputs. However, TLEs are dedicated to the use in SGP4-based simulation, and cannot directly be used as inputs for the Keplerian and circular model. Hence for the inputs of both, it is presented how they can be derived from TLEs, so one real satellite orbit is represented in all three models via *equivalent* inputs. As will be shown, this is unambiguously possible for the Keplerian orbit model. However, for the circular orbit model, it will be shown that there is no clear solution for deriving a circle's radius from a TLE. The approach chosen for this is presented later together with the implementation. (3) Lastly, the definitions of spherical and polar coordinates are fundamental to implementing the circular orbit model.

### 2.1 *space\_Veins*

For an integrated simulation of LEO satellites and mobile nodes on the ground and in the air, *space\_Veins* [4] builds upon the Connected Autonomous Vehicle (CAV) simulation framework *Veins* [11] and the internet protocol simulation framework *INET* [12]. These frameworks are based on *OMNeT++* [13] for network simulation and *SUMO* [14] for road traffic simulation.

To realize realistic satellite orbits, it employs the SGP4 model as implemented by Vallado et. al [15] in C++. It uses TLEs as input, which contain the orbital parameters, and are sourced from NORAD/NASA<sup>1</sup>. Each TLE represents one satellite

---

<sup>1</sup><https://celestrak.org/NORAD/elements/>

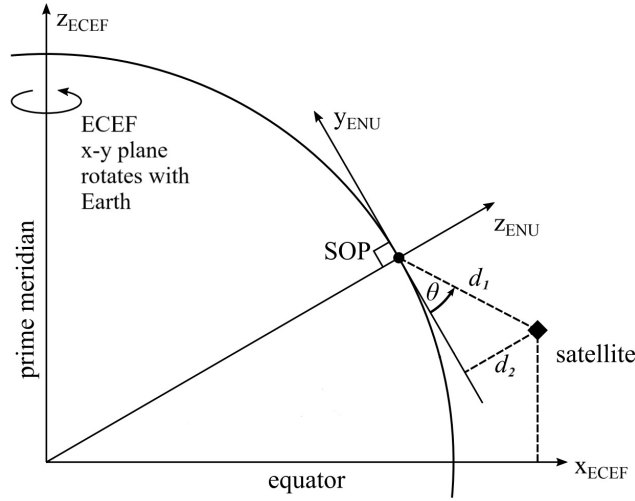
orbit. Some of the contained orbital parameters are related to the Keplerian orbit model, and are presented in the next section. Others like the drag term represent various perturbations including atmospheric drag, solar and lunar gravity, and solar radiation pressure [5] [16, Figure 2], which are not part of the Keplerian and circular orbit model. Also, Earth's non-uniform gravitation is considered, which is caused by its oblateness, i.e. non-spherical shape [9][10].

Because each TLE is derived from recent observations of a real satellite, TLEs have a timestamp, the *epoch*. As Vallado and Cefola [5] note, the epoch is "not necessarily" the time of the last observation used, so in general not only is the observation period length unknown, but also its distance to the epoch. Naturally, the SGP4-simulated orbits differ more and more from the real ones the larger the time distance to the used observations is [5]. Another important notion by Vallado and Cefola is that observed satellite maneuvers may be incorporated in a TLE, and consequently are indirectly represented when using SGP4 to simulate periods before the epoch that coincide with the observation period. However, future maneuvers are unknown to TLE creation and thus a fundamental source of errors, which vary with the maneuver magnitude.

Orbit simulation with SGP4 results in coordinates in the Earth-Centered Inertial (ECI) True Equator Mean Equinox (TEME) system. ECI systems are fixed w.r.t. to space and don't consider Earth's rotation. Franke and Sommer [4] detail why it is desirable to transform them to a Cartesian coordinate system that unites space, air and ground mobility: Firstly, Cartesian coordinate systems provide computationally inexpensive geometrical calculations for positions, speeds, and angles, the fundamental results for simulating a network. Secondly, further coordinate transformations are avoided because the air and ground segments can fundamentally be modeled in the unifying system.

For this, space\_Veins transforms the ECI TEME positions obtained from SGP4 into coordinates in a Cartesian East-North-Up (ENU) system, which is centered on a point on Earth called Satellite Observation Point (SOP). Its East-North (x-y) plane is tangential to Earth's surface at the SOP, and the Up (z) axis lies in normal direction to the East-North plane at the SOP, as displayed in Figure 2.1. In adaptation of SUMO's planar surface model, space\_Veins represents Earth's surface with this tangential plane. Before the transformation to ENU, intermediate Earth-Centered Earth-Fixed (ECEF) coordinate systems that rotate with Earth are required. More concretely, the following coordinate transformation pipeline is applied to calculate SGP4-based ENU coordinates, which is summarized in figure 2.2:

- (1) Using TLEs, SGP4 calculates satellite positions in the Cartesian ECI TEME, which uses the equatorial plane as x-y plane and Earth's polar axis as z-axis [17].
- (2) In the transformation to the first intermediate system, the International Terrestrial

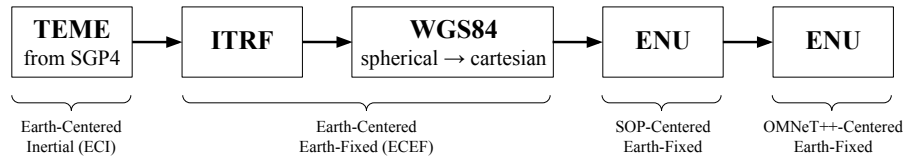


**Figure 2.1** – SOP-based transformation from ECEF to ENU coordinates (adapted from [4])

Reference Frame (ITRF), which uses algorithms adapted from *Skyfield* [18], Earth's rotation is included. From there on, the coordinates remain Earth-Fixed.

The remaining transformation steps use the *proj* library to (3) calculate *spherical* coordinates in the World Geodetic System 84 (WGS84) system, which represents positions via distance to the coordinate system center and angles w.r.t. the equatorial plane and Earth's polar axis. Spherical coordinates are explained in detail in Section 2.4. Then (4) Cartesian WGS84 coordinates are calculated and (5) Cartesian ENU coordinates w.r.t. the tangential plane centered at the SOP (6) Finally, Cartesian ENU coordinates in the OMNeT++ simulation are achieved by adding the potentially existing offset between SOP and OMNeT++ coordinate system center.

This abstraction from Earth's curvature to represent its surface introduces an error w.r.t. reality (1) in the positions of nodes on the surface and (2) at a certain altitude / z coordinate above it, except at and directly above the SOP in the center of the tangential plane. This is shown in figure 2.1, for ground positions like at one end of  $d_2$ , as well as for air nodes *created w.r.t. the surface* like at the



**Figure 2.2** – Transformations applied to coordinates of SGP4-based orbits in space\_Veins

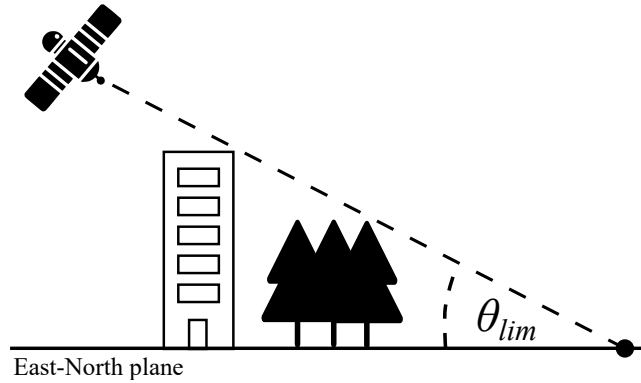
other end of  $d_2$  that use the Up coordinate/altitude  $d_2$ . For satellites, the relative position to the SOP is maintained from WGS84, i.e. no error is introduced to the distance  $d_1$ . However, when their individual ENU coordinates are considered, the Up coordinate/altitude is as erroneous as for the air nodes. In other words: Because ground positions except the SOP are erroneous, the relations of satellite positions to them are erroneous as well. An additional source for satellite position errors are the coordinate transformations to reach the WGS84 coordinate system.

In [4], Franke and Sommer evaluated the errors for ground-station-satellite relations caused by the SOP approach, analyzing distance and elevation angle, i.e. the angle between the East-North plane and the line of sight between potential communication partners. For this, they compared the SGP4-based values to values using positions calculated via Skyfield, which "works in celestial coordinates and can thus yield higher precision for an individual V2S link – at the cost of potentially sacrificing precision for in-segment links". Their results show the increasing effect of abstracting from Earth's curvature, as the distance error increases for ground stations at higher distances to the SOP w.r.t. the same satellite positions. More importantly, nevertheless their results could validate the high accuracy of the SOP approach: The relative distance error is deemed completely negligible, while the elevation angle error is said to be "negligible for most practical purposes": even for ground stations at the largest distance to the SOP, the predictions if a minimum elevation angle would be reached, and thus communication with a satellite would be available, were accurate in 99.5 % of the considered cases. space\_Veins' fundamental communication model on which this study is based is explained below.

As is evident from the space\_Veins repository<sup>2</sup> (release version 0.3), the simulations are configured via two components located in the `examples/space_veins` subdirectory: (1) A set of files representing a road network as required by SUMO, which are identified by a common prefix (in this case `Null-Island-3by3-grid`). These files (i) specify 2D road network node locations in the SUMO plane (`.nod.xml`), which is the East-North plane of space\_Veins' approach, w.r.t. a coordinate center independent of the simulated real location; (ii) edges and multi-edge routes connecting them; (iii) trips taken by simulated ground vehicles; (iv) and the relation of the SUMO coordinate system to represented locations of reality on a WGS84 ellipsoid via UTM projection (`location` element of the `.net.xml`). With `scripts/createRoadNetwork.sh` in the repository, a script to automatically generate the `.net.xml` file from a node and an edge file is provided. (2) The remaining configurations are unified in the `omnetpp.ini` file that configures the various modules of the simulation in OMNeT++, e.g. the mobility model of satellite instances.

As configured in the `omnetpp.ini` in the space\_Veins repository via the `'*.satelliteRadioMedium.typeName = "UnitDiskRadioMedium"', '*.satNic[0].radio.trans-`

<sup>2</sup>[https://github.com/veins/space\\_veins/releases](https://github.com/veins/space_veins/releases)



**Figure 2.3** – Minimum elevation angle  $\theta_{lim}$  for which satellite-ground communication is modeled to be possible.

mitter.communicationRange = 9999999km<sup>3</sup>, communication of an air/ground node with a satellite is possible when the satellite is visible to its communication partner - regardless of the distance of a visible LEO satellite, which would realistically never exceed 9 999 999 km. Modeling reality, for ground nodes and with no obstructions this means that the satellite must be *over the horizon*. Mathematically, this is represented by an *elevation angle*  $\theta$  larger than  $0^\circ$ . If potential obstructions blocking the line of sight, e.g. buildings or vegetation, shall be considered, space\_Veins allows the configuration of a higher *minimum elevation angle*  $\theta_{lim}$  as condition of communication availability. This scenario is depicted in Figure 2.3. For example,  $\theta_{lim} = 25^\circ$  is to be observed by *SpaceX Starlink* satellites per regulations [19, III-E-1 para. 42] and configured in the repository, and around  $\theta_{lim} = 63^\circ$  is used in a proof-of-concept study conducted by Franke and Sommer [4] (which is different from the accuracy study mentioned above) to consider obstructions with a height of 12 m at a ground distance of 6 m.

Another important notion of the `omnetpp.ini` is that per configuration, there is one wall-clock / real-world time at which the simulation starts ("`wall_clock_sim_start_time_utc`"). As detailed in the next section, the TLE parameter *mean anomaly*, which is related to the Keplerian *true anomaly*, specifies the position of a satellite on the orbit described by the TLE at its epoch, which is propagated to the wall-clock time of each simulation step via SGP4.

## 2.2 Keplerian orbit model

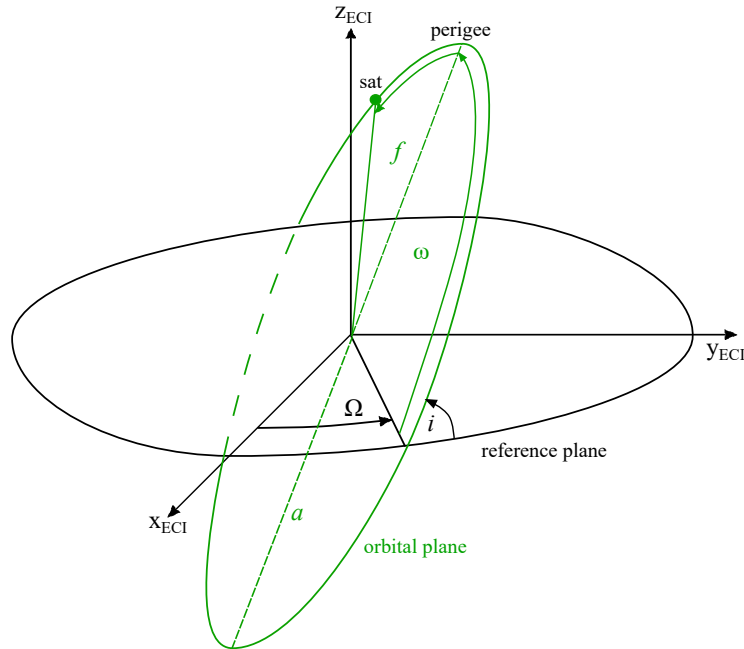
The first orbit model added to space\_Veins is based on the Keplerian two-body problem, which considers two celestial bodies orbiting each other due to the gravitation

<sup>3</sup><https://doc.omnetpp.org/inet/api-current/neddoc/inet.physicallayer.wireless.unitdisk.UnitDiskRadio.html>

of their point-like masses. I.e. all other forces, such as the perturbations included in SGP4, are ignored [6]. Accordingly, it was shown that Keplerian orbit models produce higher position errors than SGP4 [8], which will be discussed with further related work in chapter 3. For Earth orbits, satellite masses are usually considered insignificant and ignored. Thus, only the satellites move, orbiting around Earth's mass at the center of the coordinate system, which is also called one-body problem. When using an ECI coordinate system, this results in constant elliptical or circular orbits situated in an orbital plane [6].

Keplerian orbits can be specified by using 6 parameters called *orbital elements*, which are described by Lissauer and de Pater [6]. Firstly, the *eccentricity*  $e$  specifies how elliptical an orbit is, using values between 0 and 1: At  $e = 0$ , an orbit is circular, and elliptical for  $0 < e < 1$ . Values of 1 and above specify parabolic and hyperbolic movement, which, in contrast to orbits, is not bound to the central mass's influence.

The next four orbital elements specify the orbit's shape and orientation in space, and are depicted in Figure 2.4 in an ECI Cartesian coordinate system. *Inclination*  $i$  and *Right Angle of Ascending Node (RAAN)*  $\Omega$  specify the orientation of the orbital plane relative to the reference plane of the coordinate system. Commonly the equatorial plane is used as reference.  $i$  is the angle between both planes. Generally, an angle between two planes is called *dihedral angle*, and considering only the



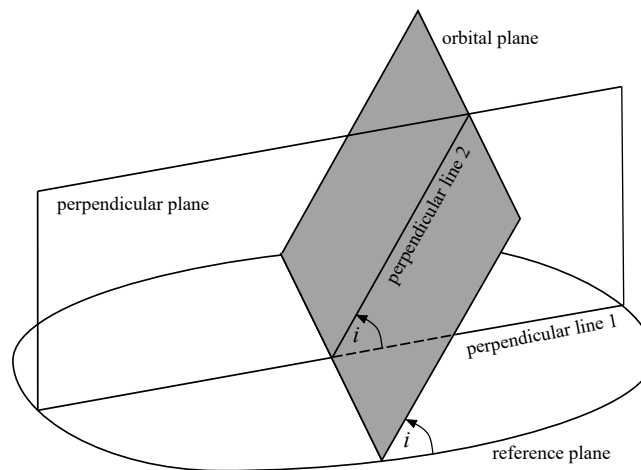
**Figure 2.4** – Five of the Keplerian orbital elements: inclination  $i$ , semi-major axis length  $a$ , Right Angle of the Ascending Node (RAAN)  $\Omega$ , Argument of Perigee / Periapsis (AOP)  $\omega$  and true anomaly  $f$

half-planes connected by it defined as "the angle between two half-lines, one in each plane, drawn perpendicular to the edge from a common point." [20], where the *edge* is the intersection line of the planes. In other words:  $i$  is the angle between the lines created by an intersection of the orbital and reference plane with a plane perpendicular to both, as depicted in Figure 2.5.  $i$  can have values from 0 to 180°.  $\Omega$  is the angle to the coordinate systems reference direction (x) at which the orbit passes the reference frame from below to above, ranging from 0 to 360°, excluding 360° itself. At the other end of the intersection line, 180 additional degrees, it passes below again. Thus  $\Omega$  also describes the orientation of the intersection line. The position where the orbit is closest to Earth is called perigee, the farthest apogee. The line connecting them is called the semi-major axis, and its length  $a$  is the fourth orbital parameter. It essentially specifies the orbit's size for a given eccentricity. With the *argument of periapsis/perigee* (AOP)  $\omega$ , the angle between ascending node and perigee ranging from 0 to 360°, excluding 360° itself, the orbit's orientation in the orbital plane is specified.

The last parameter *true anomaly*  $f$  specifies an instantaneous position on the orbit via the angle between its direction and the perigee direction, ranging from 0 to 360°, excluding 360°, and is also included in figure 2.4.

Because the orbital elements describe orbit shape and orientation in a relatively intuitive way, they can be used to create predictable artificial orbits, without requiring TLEs as reference, relatively simply.

From these 6 orbital elements results an orbit with velocities characterized by formula 2.1 given by Lissauer and de Pater [6], which uses the radius  $r$  to denote



**Figure 2.5** – Definition of inclination via two lines perpendicular to the reference-orbital plane intersection line; in a plane perpendicular to reference plane and orbital plane

the distance to the coordinate system center, and  $GM$ , the standard gravitational parameter of Earth in  $\text{m}^3/\text{s}^2$ :

$$v = GM \left( \frac{2}{r} - \frac{1}{a} \right) \quad (2.1)$$

This means that velocity decreases with increasing radius. Thus elliptical orbits have varying velocities, while velocity is constant for circular orbits.

#### *Relation to TLE parameters*

To use one TLE as input both for SGP4 and the Keplerian orbit model, the 6 orbital elements can be unambiguously derived from it. This is because firstly, a TLE contains [5, Figure 1] the four orbital elements eccentricity, inclination, RAAN and Argument of Periapsis (AOP). Secondly, it also contains the mean anomaly from which the true anomaly can be calculated [21]. As this can be realized via library functions, the underlying mathematics are beyond the scope of this thesis. Lastly, the semi-major axis length can be calculated by combining formulas 2.2 and 2.3 as given by Lissauer and De Pater [6], resulting in formula 2.4. Here,  $n$  is the *mean motion*, which needs to use the unit *revolutions/s* to be directly compatible with the  $GM$ .

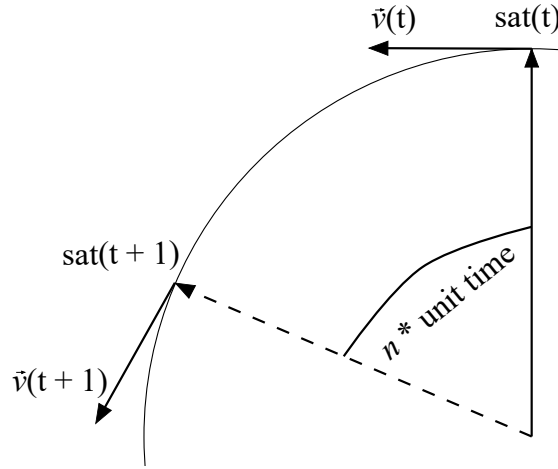
$$P_{orb}^2 = \frac{4\pi a^3}{GM}, \quad (2.2)$$

$$n \equiv \frac{2\pi}{P_{orb}}, \quad (2.3)$$

$$a = \left( \frac{GM\pi}{n^2} \right)^{-3} \quad (2.4)$$

The mean motion is also part of a TLE. However, there the unit *revolutions/day* is used. Mean motion is a measure of the speed of progress in the orbit, as it is the mean angle between the vector of a satellite position at one point of time and the position vector of the next unit time step, as depicted in Figure 2.6. It is important to note that in contrast to mean motion, the velocity  $v$  is a more abstract measure of progress speed on the orbit: It is a vector at each satellite position that points in the direction where the satellite will move at this moment (Figure 2.6) - hence it is different for every point. Constant progress speed in circular orbits is characterized by velocity vectors with the same length, i.e. a constant amount of velocity. This makes it clear that the  $v$  in formula 2.1 represents such a velocity amount.





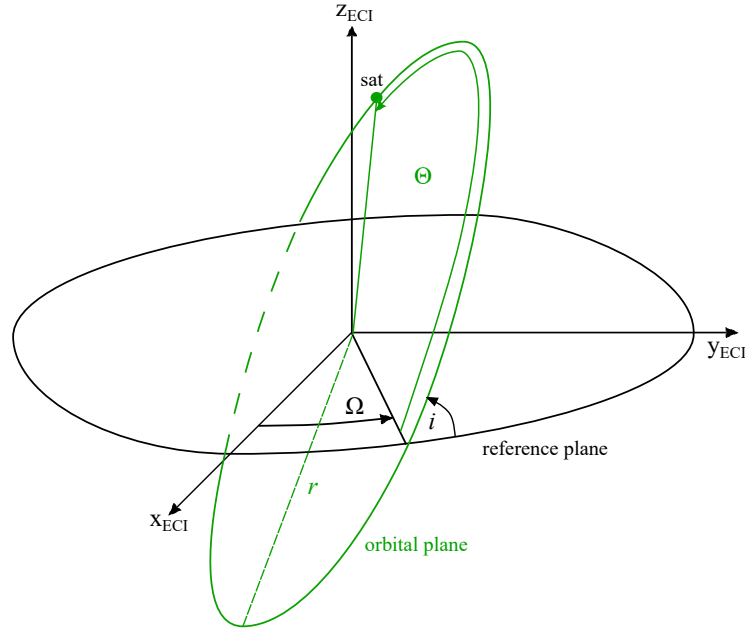
**Figure 2.6** – Satellite position at moment  $t$  and the position predicted by the mean motion  $n$  at  $t + 1$ , and their velocity vectors

## 2.3 Circular orbit model

As described in the previous section, constant (w.r.t. space) circular orbits are a special case of the Keplerian one-body problem. The second orbit model implemented in thesis approximates all orbits as circular ones.

In this case, no eccentricity is needed to describe the orbits. Also, the radius  $r$  can serve as a more intuitive measure for the orbit's size instead of the semi-major axis length  $a$ . Because all orbit points are equidistant to the coordinate system center, a semi-major axis through the center can be oriented arbitrarily, and always has a length of  $a = 2r$ . In other words, circular orbits do not have a distinct perigee. For this reason, no AOP is needed to describe the orbit's orientation in the orbital plane. However, without it, the true anomaly loses its reference angle, and cannot be used to describe instantaneous satellite positions. Instead, an angle in the orbital plane, from the ascending node to the satellite position, can be used, as presented by Mathavaraj and Padhi [22]. As it adapts the idea of the true anomaly - angular progress on the orbit - it is called *circular anomaly*  $\Theta$  in this thesis. Lastly, the orbital elements inclination  $i$  and RAAP  $\Omega$  describing the orbit plane's orientation w.r.t. the (equatorial) reference plane remain unchanged. This adaptation of the Keplerian model is presented in Figure 2.7.

Because the circular orbit model adapts and even simplifies the Keplerian orbital elements, it provides the same relative ease and intuitivity of creating predictable artificial orbits as the Keplerian model. These artificial orbits obviously are limited to be circular, even though the model can model real elliptical orbits as described below.



**Figure 2.7** – The circular orbital elements: inclination  $i$ , radius  $r$ , Right Angle of the Ascending Node (RAAN)  $\Omega$ , circular anomaly  $\Theta$

#### *Relation to TLE parameters*

To use one TLE as input for both SGP4 and the circular orbit model, 4 of the 5 circular parameters can be derived from a TLE unambiguously:

Similar to the Keplerian orbit model, inclination and RAAN from a TLE can directly be used. The circular anomaly can, per definition, be derived by the addition of a TLE's AOP and true anomaly. Recalling Formula 2.1, velocity in Keplerian circular orbits is constant. Thus, a circular orbit's mean motion is equal to its actual angular motion. Consequently, the mean motion from a TLE can be used to specify how the circular anomaly  $\Theta$  progresses in its circular approximation.

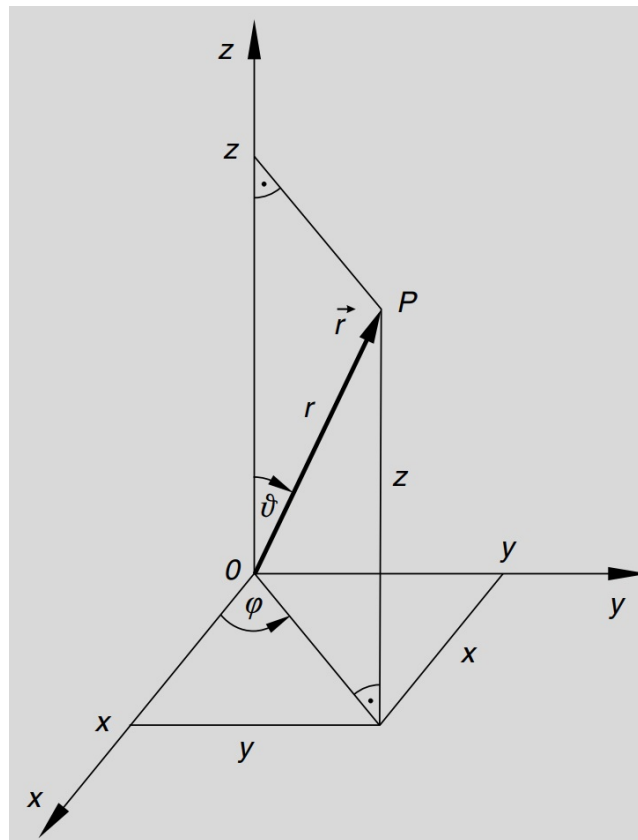
However, there is no clear way to derive an optimal radius for a circular approximation of an elliptical orbit from a TLE. This is because for elliptical orbits, the semi-major axis length  $a \neq 2r$ , as there are different radii. In this thesis, this gap is closed by calculating the radius parameter from SGP4 simulation of the TLEs, which is presented in detail in chapter 4.

## 2.4 Spherical and polar coordinates

The approach applied in this thesis to implement the circular orbit model is based on spherical and polar coordinates. Spherical coordinates can be used alternatively to Cartesian coordinates to describe positions in three-dimensional space, using two

angles and a distance to the coordinate system center. The following explanations and equations are based on the work by Papula [23, pages 124-127]. The spherical coordinates and their relation to the Cartesian axes are depicted in Figure 2.8.

For a point  $P$ , the azimuth angle  $\Phi$  is the counterclockwise angle between the  $x$ -axis and the direction of the vector from the coordinate center to  $P$  in the  $x$ - $y$  plane, ranging from  $0^\circ$  to  $360^\circ$ , excluding  $360^\circ$ . The polar angle  $\rho$  is the angle between the positive section of the  $z$ -axis and the vector to  $P$  from the coordinate center, in the direction of the negative section of the  $z$ -axis. In the plane in which both the  $z$ -axis and the point vector lie, one can generally find two angles from the  $z$ -axis to the point vector which together make up  $360^\circ$ . Of those, the polar angle is the smaller one in principle, ranging from  $0^\circ$  to  $180^\circ$ . When  $P$  is on the negative section of the  $z$ -axis, both angles measured from  $+z$  in  $-z$  direction are  $180^\circ$ , so in this case both are the polar angle. The third dimension of spherical coordinates is the radius  $r$ , the distance from the coordinate system center to  $P$ . Cartesian coordinates are related to spherical coordinates via the following equations:



**Figure 2.8** – Spherical coordinates: Azimuth angle  $\Phi$ , polar angle  $\rho$  and radius  $r$  for the point  $P$ , and their relation to the Cartesian axes [23].

$$x = r \cdot \sin \rho \cdot \cos \Phi, \quad (2.5)$$

$$y = r \cdot \sin \rho \cdot \sin \Phi, \quad (2.6)$$

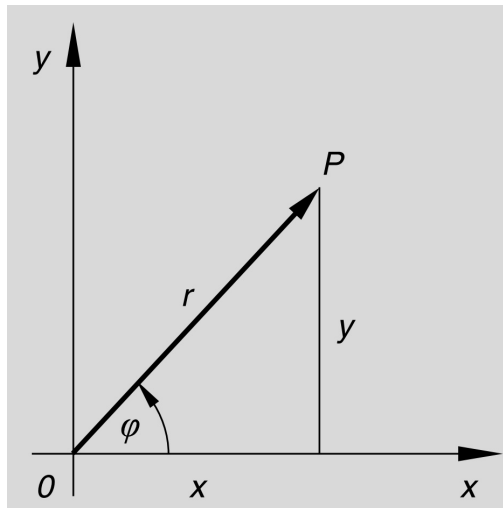
$$z = r \cdot \cos \rho \quad (2.7)$$

Similarly, the azimuth angle and the radius can be used instead of Cartesian coordinates in two dimensions [23, pages 94-95]. For a point  $P$ , in this case,  $\Phi$  too is the counterclockwise angle between the x-axis and the vector to the point, and  $r$  too is the distance from the coordinate system center to this point. This is depicted in Figure 2.9.

Cartesian 2D coordinates are related to spherical coordinates via the following equations [23]:

$$x = r \cdot \cos \Phi, \quad (2.8)$$

$$y = r \cdot \sin \Phi \quad (2.9)$$



**Figure 2.9** – Polar coordinates: Azimuth angle  $\Phi$  and radius  $r$  for the point  $P$ , and their relation to the Cartesian axes [23].

---

## Chapter 3

### Related Work

---

Various models for satellite orbits exist in related work, with different approaches to realize and evaluate them. Of particular interest for this thesis are evaluations of their effects on communication metrics, especially for individual ground-satellite links, like propagation delay and satellite visibility. This is because under the assumption that the positions and movements of involved ground stations remain unchanged, differences in these low-level metrics can directly be related to the differences in satellite positions caused by different models. This is contrasted by the evaluation of more complex communication, like end-to-end communication between ground stations via one or more inter-satellite links. Also in this case, different satellite positions are the cause of different communication metrics. However, relating positional and communication differences is complicated by the fact that observed effects on communication are a superimposition of the differences caused for individual links. To study the relation between differences in position and communication, evaluations of the positional effects of different models are highly relevant as well. In the following, firstly Space-Air-Ground Integrated Networks (SAGINs) and space-ground-integrating simulators are analyzed on which orbit models they employ and how they are evaluated. This is followed by works that focus on the evaluation of the circular, Keplerian and SGP4 orbit models in the described aspects.

Firstly, works that present SAGIN simulators like `space_Veins` are discussed and analyzed in those aspects. Franke and Sommer [4] conducted a review of such works, and only found two which model space and ground mobility in detail. The SAGIN simulator presented by Cheng et. al [24] simulates LEO satellites by generating traces with the closed source simulator Systems Tool Kit (STK)<sup>4</sup>. These pre-computed orbits are then transformed into files for their simulator. However, STK offers various models for orbit simulation<sup>5</sup>, and the work does not specify which one is used.

---

<sup>4</sup><https://www.ansys.com/de-de/products/missions/ansys-stk>

<sup>5</sup>[https://help.agi.com/stk/index.htm#stk/vehSat\\_orbitProp\\_choose.htm](https://help.agi.com/stk/index.htm#stk/vehSat_orbitProp_choose.htm)

Only a case study to validate the usefulness of the simulator is performed, but no evaluation of its accuracy or the role of the satellite mobility model in particular. The other simulator closely related to space\_Veins is presented by Puttonen et al. [25] and deals with system-level simulation of 5G non-terrestrial networks. While they implemented multiple satellite models directly in their simulator, only one includes mobility. For this, they used a C++ implementation of SGP4, which is no longer available under the reference they provided. They perform verification of their simulator's compliance with specifications by the 3rd Generation Partnership Project based on a comparison of higher-level communication metrics in various scenarios, which does not entail evaluation of positional accuracy or propagation delay and availability.

Additionally, Franke and Sommer identified simulators that focus on detailed models of satellite mobility, but use more abstract models for ground mobility. Fraire et. al [26] propose the FLORASAT simulator for Direct-to-Satellite Internet of Things (IoT) systems, where LEO satellites communicate with IoT devices. This work also leverages STK to pre-compute satellite traces, without specifying which of the available orbit models are used. To validate their simulator, they perform simulation of a benchmark scenario for which higher-level network and protocol metrics are evaluated, such as the packet drop rate. The open source ESTNeT proposed by Freimann et. al [27] employs two mobility models: a purely Keplerian without perturbations, and SGP4. Also, they model satellite maneuvers, which are not included in the Keplerian and SGP4 model. They demonstrate the capabilities of their simulator by experiments on e.g. solar panel power generation and antenna power consumption, but do not validate it w.r.t. positional accuracy or propagation delay and availability behavior. Valentine and Parisi [28] present an open source simulator that integrates INET and the Open Source Satellite Simulator (*OS<sup>3</sup>*) leveraging SGP4. To remove the dependency on TLE files from NASA/NORAD, this work also implements another mobility model based on orbital parameters, which are made up of Keplerian elements and other, not fully specified, characteristics. The simulator is validated by comparing Round Trip Times (RTTs) for ground-satellite-ground communication with benchmark data. Wang et. al [29] propose a cloud-based simulation platform for space-ground integrated networks, which also uses STK without providing details on the exact satellite mobility model. Additionally, they highlight that STK enables the simulation of satellite maneuvers. They validate their simulator's capabilities by experiments studying the metrics of inter-satellite links. Kassing et. al [30] propose the HYPATIA simulation framework, which implements satellite mobility via SGP4. This is another work that not only uses TLEs sourced from NASA/NORAD, but also generates TLEs from Keplerian orbital elements. Without further details, it is claimed that the SGP4-based model "adds a 1-3 km error per day to satellite trajectories". They apply their framework to evaluate network metrics of end-to-end

communication between ground stations using inter-satellite links, e.g. RTT and bandwidth, but do not validate the simulation framework itself. The work by Schubert et. al [31] presents *ns-3-leo* for LEO satellite mobility and channel simulation. It employs SGP4 to simulate existing satellites, and a purely circular orbit model for others, for which is claimed that “it still is precise enough to accurately describe the satellites’ positions within a network topology.”. However, they only validate the positional accuracy of their approach via a global visualization of a constellation. Also, they evaluate runtime and memory performance, and apply their simulation to evaluation of network protocols for metrics like end-to-end delay between ground stations using inter-satellite links. The LEO-ground virtual testbed CELESTIAL [32] also uses SGP4, and evaluates ground-satellite-ground end-to-end network metrics like delay and jitter in a case study. Another LEO-ground simulator is presented by Kempton and Riedl [33], and is the second one of the works reviewed by Franke and Sommer that simulates satellites based on Kepler’s two-body problem. They claim that their implementation “offers a good balance of accuracy and low computation requirements.”, without backing this up with further details. Instead, a case study using the simulator is done to evaluate the effects of different satellite networks on ground-satellite-ground path latency.

An earlier review of satellite-terrestrial integrated network simulators was performed by Jiang et. al [34]. Of the simulators capable of satellite orbit simulation identified by them [34, Table 6], three works not analyzed by Franke and Sommer were accessible for this thesis: The first work by Zhang et. al [35] does not focus solely on LEO satellites, but also includes such on middle-earth and geostationary orbits, and is another work using STK to simulate them, without specifying the used orbit model. They evaluate the multi-orbit-type constellation designed in their work and network algorithms, but neither positional accuracy nor that of network metrics. Liu et. al propose a simulator that focuses purely on satellite mobility and inter-satellite communication [36], also combining LEO, middle-earth orbit and geostationary satellites. It simulates the orbits based on Kepler’s two-body problem. The work does not evaluate accuracy, but only metrics of inter-satellite communication and computational costs of the simulation. Finally, Petrosino et. al [37] simulate the communication of LEO satellites to NarrowBand IoT devices on the ground. They use a satellite mobility abstracted to movement on the Cartesian x-axis. The simulator is used to evaluate the performance of a specific NarrowBand-IoT satellite-based communication system.

The above survey of satellite simulation in space-(air-)ground integrated simulators shows that of the orbit models considered in this thesis, SGP4 is the most popular, the Keplerian model is used in three cases, and the circular model in only one case. Also, two of the works propose another approach to simulating non-existing satellites

than using other models than SGP4, which however is not further explored in the thesis: TLEs are created artificially. Positional accuracy is only evaluated in one work for SGP4 [30], which “adds a 1-3 km error per day to satellite trajectories“, and the effects of the models on individual ground-satellite links is evaluated in none of those works. Hence, analysis of works beyond space-(air-)ground integrated simulation is required.

Research for this thesis yielded three works that evaluated orbit models based on Kepler’s two-body problem: Thammawichai and Luangwilai [8] compare a non-linear programming approach for LEO orbit prediction with SGP4 and the purely Keplerian model. They state TLEs are not only used as SGP4 input, but also as “pseudo-observations” as basis of position comparison with SGP4, which is neither clarified in the work nor in the according reference therein. Hence, it is unclear which ground truth is used. Some works discussed in the remainder of this section compare the results calculated from recent TLEs with older TLEs, so possibly this is what is meant by the authors. Using the SGP4 implementation by Vallado and Crawford [38], position error results are presented for 5 satellites for a propagation period of 10 to 20 days. From the presented data, it is not possible to exactly quantify the errors, but it is clear that in comparison the Keplerian model performs worse. It consistently reaches errors of over 2000 km after 6 days of propagation, whereas the SGP4 error remains below 200 km in most cases. In the one exception, the Keplerian error is still roughly 3000 km larger than the SGP4 one at the same propagation time. Also in the first few days of propagation, the Keplerian model performs worse. In the work by Morales et. al [39], SGP4 and a purely Keplerian model for LEO satellites are compared, along with a Kalman-filter-based method. Satellite positions based on GPS are used as ground truth. For SGP4, the authors state that at the TLE epoch, already a position error of around 3 km exists, which over the propagation period of “approximately 450 seconds” increases by 600 m for one satellite, and decreases by roughly 225 m for another. As the propagation using the Keplerian model is initialized from a truth position, its error starts at 0 m, but increases much faster than the SGP4 one, reaching 1000 m in the 450 s propagation period for the first satellite, and around 1100 m for the other satellite. While Biria [40] evaluates Keplerian orbit propagation, it is only w.r.t. runtime, not positional accuracy or runtime performance.

For the Keplerian model one thus can conclude that it reaches positional errors of 1000 km and above after a few days, and grows at least one kilometer in 450 s. As the work by Morales et. al [39] providing short-term insights uses a Monte-Carlo simulation, where in each step the next position is calculated from a truth position, the error without using intermediate truths may be higher than that. This shows that additional research on the short-term positional errors of a purely Keplerian model is



required, and that there is a research gap for their influence on the communication metrics of individual ground-satellite links.

Only one work could be identified that assesses the accuracy of a circular orbit model. Kast [41] compares circular orbit and Fourier power series ephemeris representations with a reference ephemeris predicted by the Goddard Trajectory Determination System, which provides a similar accuracy as precise orbit determination techniques [42], which are listed by Vetter [43, Table 1] with errors below 200 m. The circular model is fit to the reference data for 3- and 7-day periods. The study evaluates the components of positional error, i.e. along-track/in-track, cross-track and radial errors. Along-track/in-track errors refer to the errors in the direction of the satellite's velocity vector, cross-track to errors perpendicular to the velocity vector, and radial errors are of the direction of the radius vector [44]. The results show that for both fit periods and a prediction period of one day, the maximum positional error is dominated by along-track error at 30 km and radial error at 10 km. When the prediction period and fit period are 7 days, only the cross-track errors increase considerably, to roughly 10 km. When only a 3-day fit is used for 7-day prediction, the along-track error increases to 65 km. However, this does not provide general insights for a circular model, but only one that is fit to an ephemeris via Kast's methods. Consequently, there also is a research gap for positional errors of circular orbit models and their effects on the ground-satellite link communication metrics, even though Kast's results show what a circular model can potentially achieve.

In contrast to the Keplerian and circular orbit model, there are plenty of works which evaluate the accuracy of SGP4. Thus one approach, which is applied in this thesis, is to evaluate the Keplerian and circular model by assessing their errors w.r.t. SGP4, and use the existing evaluations of SGP4 to assess their general performance. One important consideration in analyzing these works is which SGP4 implementation is used, as Vallado et. al [15] highlight that different implementations can differ in their results by 10 km and more in the worst case.

The first group of works analyzing SGP4 considers the total position errors of predicted orbits. Baghel et. al [45] performed a long-term comparison of the results of older TLEs with the ones of more recent TLEs, which shows the deterioration of their accuracy over a maximum period of 10 days. For this, they implemented the code given by Vallado et. al [15]. Both the evaluated HySIS satellite [46] and the KALAMSAT satellite are LEO satellites, which for the latter could only be confirmed by an online article<sup>6</sup>. One interesting result is that the errors vary considerably for the two satellites they considered, especially for a higher TLE age: The average error of comparisons using 10-days-old TLEs of the second satellite is almost twice as big as the average error of the other satellite. The shortest period for which they

<sup>6</sup><https://www.bbc.co.uk/news/world-asia-india-46956595>

provide numerical results is a TLE age of two days, for which one satellite has a positional error of roughly 2 km, the other one around 1.1 km. This work however only shows how older TLEs compare to recent ones, and does not use ground truth data which describes a satellite as accurately as possible. Haidar-Ahmad et. al [47] use a LEO satellite's true ephemeris generated by a satellite's GPS receiver as ground truth. Their work improves upon an unspecified implementation of SGP4 via Machine Learning and evaluates the improvement of position error for the first and second pass over the location of Irvine in California. Unfortunately, the work does not specify the time distance between the TLE epoch and the first pass, when the positional error of SGP4 is around 1 km. However, the time distance to the second pass is given, from which can be derived that the SGP4 error increases about 6 km over a time of 1.6 hours. In the work by Khairallah and Kassas [9], which improves upon SGP4 accuracy by applying satellite tracking information, positions calculated by an unspecified SGP4 implementation are also compared against GPS-based truth ephemerides. Their experiments result in a position error that is constantly around 2 km for a duration of 300 seconds starting at the TLE epoch. Vallado and Crawford [38] examine the difference in position error of SGP4 applied to time periods of the observations which are used to generate a TLE, and applied to the following time periods, based on their own SGP4 implementation, which is commonly cited in the previous works; either as [38] or [15]. Also space\_Veins bases its implementation on the latter [4]. Regarding the position error of LEO satellites, they considered an ICESat ephemeris [48], using truth "generally accurate to a few cm ". During the period to which the TLE was fitted, it produced errors  $< 1.5$  km. In the following time, errors vary in this range for about 1 day, after which there are increasingly larger errors up to 2.5 km after another 2/3 of a day. Additionally, the position errors show a periodic behavior, which is not explained in this work. An explanation is given by Easthope [49], who performs an analysis of the mathematics of SGP4 in comparison to another analytical solution including various perturbations: A term which he calls "primary harmonic term" is not included in SGP4, which is why errors in a period that aligns with the orbital period occur. Wei and Zhao [50] employ the SGP4 implementation by Vallado et. al to evaluate the position error of Low-Earth and other orbits, comparing SGP4 results with a precision track model which includes more perturbations than SGP4. For the LEO orbits and a propagation period of one day, this yields errors ranging from 3 to 10 km, increasing to 10 to 40 km for 3 days, up to 50 to 1000 km for 15 days. The error increases with higher orbit altitudes.

The other group of works considers the components of positional error, i.e. along-track/in-track, cross-track and radial errors as explained above for Kath's work that analyzed a circular orbit model. For this perspective, Kelso [51] provided the fundamental result that the along-track error is the vastly dominant component.

Hence, its magnitude is approximately comparable to the total position errors in the previous works. Kelso's evaluates not LEO, but GPS satellites, and uses ground truth ephemerides by GPS providers. An STK implementation of SGP4 that is based on Vallado's implementation is used. Firstly, his long-term evaluation for a period of 15 days after and before the TLE epochs shows that calculations backwards in time are much more accurate, and even perform better for the first few days before than at TLE epoch, after which they deteriorate again. Similar to the works above, which did not consider backward calculations, accuracy steadily worsens with increasing forward time distance. Secondly it is shown that an error of a few kilometers is already present at the TLE epoch. As this work did not evaluate LEO satellites, absolute numbers are omitted here. An important conclusion by Kelso is that "Error characteristics for satellites in similar orbits can be considerably different. As such, the error characteristics of each satellite should be determined independently." Also Kelso notes that SGP4 positional errors show periodic behavior. Wang et. al [52] examine the positional error components for LEO satellites of different eccentricities. They do not specify which SGP4 implementation is used, and use positions calculated from a TLE at a given epoch in comparison to the results of older TLEs. Firstly, their results agree with Kelso's findings that the in-track (here downrange) error dominates the total error. Secondly, for a propagation period of one day, this yields presumably averaged in-track errors of around 3 km at maximum for orbits with an eccentricity of 0.02 or smaller, and a perigee altitude of 400-500 km. For the same eccentricity and higher perigee altitudes, all other in-track errors are lower, ranging from around 0.08 to around 1.32 km. For orbits with higher eccentricities, the error *mostly* increases, up to around 8 km for eccentricities between 0.2 and 0.7 at an altitude of 800-900 km. However, no linear relation between altitude and downrange error or eccentricity and downrange error could be shown. Wesam et. al [53] performed an error analysis for a LEO satellite in which they compare an orbit calculated with STK using the high-precision orbital propagator with the SGP4-calculated orbit based on a TLE created by them to fit the STK-based ephemeris. They do not specify which SGP4 implementation is used. No numerical results are provided for short propagation periods like a day, but it is shown that the in-track error, which is also clearly dominating in this work, can reach 30 km after 5 days. However, in another evaluated case, it is only around 10 km after 7 days.

Thus, the works analyzing positional errors by SGP4 can be summarized by the fact that there is no general behavior of the error for certain eccentricities, altitudes or time periods. The works only show that the error increases with longer propagation durations, and altogether confirm Kelso's finding that "error characteristics for satellites in similar orbits can be considerably different." However, from the assessments of a short propagation period of a day starting at the TLE epoch (in contrast to the

work of Haidar-Ahmad et. al [47]), it can be estimated that the positional error is about 10 km at maximum. As only Kelso [51] considered backward propagation from the TLE epoch, and for much longer periods, in the absence of more precise backward knowledge this estimate is applied for forward and backward propagation. However, his findings for better accuracy in backward propagation indicate that this is more considerable overestimation in this direction. As not all works make it clear which SGP4 implementation is used, this estimate is increased to 20 km following Vallado's findings on different results by different implementations. Also, an initial error of a few kilometers at the TLE epoch exists, e.g. 2 km in the results of Khairallah and Kassas [9].

For the evaluation of the Keplerian and circular model for propagation periods of less than a day by comparing them to SGP4, this means that the comparative positional error can be related to real orbits by considering the maximum 20 km SGP4 error estimate. However, because of the lack of assessment of SGP4's effects on communication metrics of individual satellite-ground links, the comparison of these metrics in the Keplerian and circular model to SGP4 cannot be related to real orbits. In other words, this approach is only able to evaluate how the use of the alternative models changes the communication characteristics in space\_Veins, but cannot assess in which communication deviations to reality they result.

---

## Chapter 4

# Implementation

---

Based on the theoretical descriptions of the orbit model alternatives to SGP4 in chapter 2, they were implemented in `space_Veins` in the following ways.

### 4.1 Keplerian orbit model

As the mathematical implementation of the Keplerian model along the lines of the laws by Kepler and Newton [6] would require thorough validation, it was decided to use a validated library instead. For this, the Python library *poliastro* (version 0.17) was chosen, as it provides an easy-to-use high-level API and is “the longest-lived Python library for astrodynamics” used by various researchers and enterprises [54]. It is open-source and purely based on Python. Besides the Kepler problem, it offers implementations of e.g. other orbit propagation methods and maneuver calculations, and also provides quick visualization of e.g. Earth orbits.

To enable the use of a Python library with the C++ - based `space_Veins`, the following method using orbit traces, i.e. precomputed coordinates, was implemented:

(1) Keplerian orbital elements are parsed and calculated from TLEs. For calculating the semi-major axis, equation 2.4 was implemented. For calculating the true anomaly from a TLE’s mean anomaly, *poliastro*’s functions `M_to_E` and `E_to_nu` were chained, using eccentric anomaly (described by Battin [21]) as an intermediate step. This was compared to the semi-major axis and true anomaly calculation by the library used for TLE parsing *TLE-tools*<sup>7</sup> (version 0.2.3). Their results were equal, so *TLE-tools* was chosen for code simplification.

(2) The TLE-based Keplerian elements are used with *poliastro*’s simulation of the simplified two-body problem, i.e. one-body problem, to generate orbit traces for a desired duration, as was done in some of the satellite simulators discussed in the previous section. This is done with the function `twobody.Orbit.from_classical`,

---

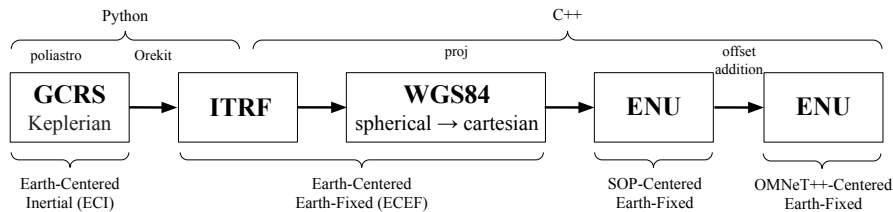
<sup>7</sup><https://pypi.org/project/TLE-tools/>

which takes the Keplerian elements and creates an `Orbit` object. Once again, `TLE-tools`, which also uses `poliastro`, offers an implementation of this. It was tested against a manual implementation, resulting in equal `Orbit` objects; hence the choice was made to rely on `TLE-tools` for this step too. From `Orbit` objects, a list of 3D coordinates is created using their `to_ephem` function, which is parameterized by a start time, the duration and the time step in which positions from the orbit are sampled. To calculate traces for the simulation, this maps to the simulation start time, duration and time step in which satellite positions are updated. Similar usage of the `Orbit` and `Ephem` class are presented by Rodríguez and Garrido [54]. The coordinates produced by `poliastro` are in the ECI Geocentric Celestial Reference System (GCRS). Even though TLE parameters are meant to describe orbits in the ECI system TEME, figure 4.2, which is further explained below, and the results in section 5.6 show that they seem to be compatible with GCRS, although `poliastro` is not aware of the relation of its inputs to TLEs.

(3) As `proj`, which is used for many coordinate transformations in `space_Veins`, does not support GCRS, another step is done in Python: GCRS coordinates are transformed to the ECEF ITRF, for which a transformation pipeline to ENU already exists in `space_Veins`. Two libraries for this direct transformation were compared: `astropy` [55] (version 5.3.4), which is also used by `poliastro`, and `Orekit`<sup>8</sup>, a Java library with a Python wrapper. For all tested constellations, for which traces were created for periods of 3 and 6 hours, `Orekit` achieved a speedup of the whole trace creation of larger than two, while the average distances to equivalent ITRF coordinates from SGP4 increased by only less than a meter, which is marginal w.r.t. SGP4's position errors discussed in the previous section. Hence, `Orekit` was selected for the GCRS-to-ITRF transformation.

For each satellite / TLE, one ITRF trace file is created. It is then continuously read during simulation, one entry per simulation time step. These coordinates can then be converted from ITRF to ENU coordinates using the same steps as for the SGP4-derived coordinates. This adaptation is summarized in Figure 4.1. The implementation of this adapts `space_Veins`' `SGP4Mobility` class responsible for

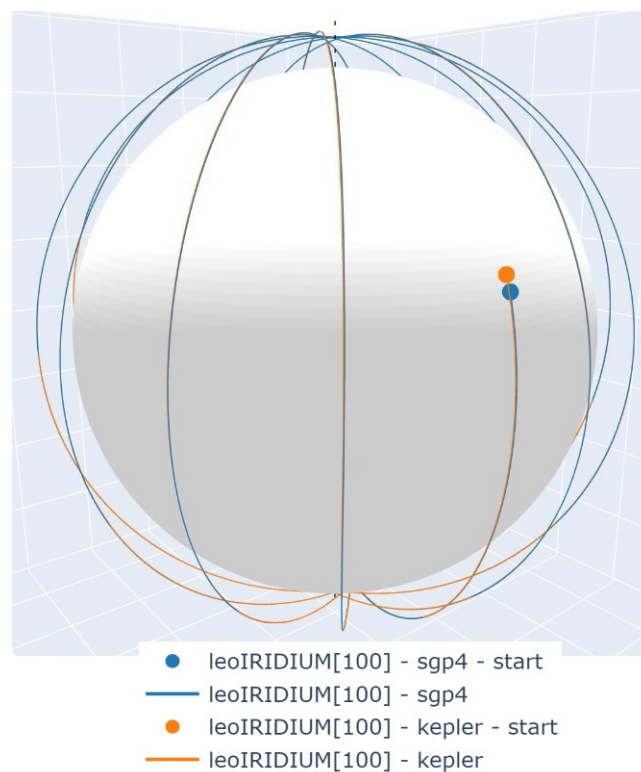
<sup>8</sup><https://www.orekit.org/>



**Figure 4.1** – Transformations applied to coordinates of Keplerian orbits created in `poliastro`

the movement of SGP4-based satellites to the new `KeplerMobility` class by (1) replacing initialization via TLE parsing with an initialization that opens the trace file, and (2) replacing the calculation and transformation of TEME coordinates to ITRF with reads from the trace file. The following calculations remain unchanged.

Accordingly, the `SatelliteInserter` class responsible for handling the satellite instances is extended with an OMNeT++ parameter `pathToTracesDir` that specifies the directory containing the trace files of the current configuration, and functions that create a satellite instance for each trace file. In contrast to this, the existing `SatelliteInserter` implementation creates satellite instances from a file with a list of TLEs. Also, a `mobilityType` parameter has been added to the `SatelliteInserter` to specify if the functions for creating satellite modules



**Figure 4.2** – Visualization of the orbits of the IRIIDIUM-100 satellite in the SGP4 and Keplerian model in the ECEF coordinate system ITRF, the first coordinate system of the transformation pipelines that is used for both models, simulated for 12 hours. In contrast to an ECI perspective, the orbits rotate with Earth. Even though at the same start time, the position is clearly different, the trajectories from there on are so similar on this scale that they seem to overlap, sometimes showing the Keplerian orbit, sometimes the SGP4 one - e.g. on the very left. Because in this instance, the simulation start wall-clock time is at a quite high distance to the epoch, there is a significant difference between the starting positions.

with the SGP4, Keplerian or circular orbit model are to be used, which for the Keplerian model needs to be set to "Kepler". The mobility model needs to be specified for the satellite modules' mobility submodule too, using the already existing `mobility.type` parameter. It needs to be set to the used mobility class, i.e. here to "KeplerMobility" instead of "SGP4Mobility".

This approach has altogether been validated by: (1) the comparison of the TLE-tools library with an implementation of equation 2.4 and `poliastro`'s true anomaly calculation, showing that one can rely on TLE-tools for a correct derivation of orbital elements and the creation of `poliastro Orbit` instances, (2) the comparison of `astropy` and `Orekit` for GCRS-to-ITRF transformation, which with its very similar coordinate results showed the correct application of the transformation in either library, and (3) the resulting orbits being reasonably similar to the SGP4-based ones, as examined in detail in section 5.6. This very close similarity on a macro scale, which is presented in Figure 4.2, showed that the TLE parameters were most likely correctly adapted to the Keplerian model. This also shows that the parameters originally related to the TEME system are seemingly compatible with GCRS.

## 4.2 Circular orbit model

As described in section 2.3, it is possible to treat the circular orbit model as a special case of the Keplerian one, and thus use the `poliastro`- and trace-based approach from the previous section. Because of the general computational efficiency of C++ in contrast to Python, it was decided to fully implement the circular orbit model directly in the `space_Veins` code instead. To also use TLEs as input, section 2.3 already related inclination, RAAN and circular anomaly to TLE parameters. Additionally, the implementation requires (1) determination of the radius of the circle, as elliptical orbits have different distances to the Earth's center and the relation  $r = \frac{a}{2}$  to the derivable semi-major axis does not hold. (2) Also, algorithms that calculate a satellite's position at a certain time from inclination, RAAN, circular anomaly and radius are required. The following solutions were developed for those problems:

(1) As SGP4 achieves relatively small distance errors, as shown by the related work, and is the reference model of this thesis, an approach to derive a circular radius from the SGP4 simulation of the same satellite is taken. A first implementation of this used the TLE to calculate the SGP4 position at the simulation start time, of which the distance to the coordinate system center determined the radius for the equivalent circular orbit. This was done in the earth-centered SGP4 result coordinate system TEME. The distance error w.r.t. SGP4 could be reduced in a second version, where, before running the simulations in the circular model, a simulation of the same desired duration in SGP4 is done. From its results, the average distance of



a satellite to the coordinate system center is calculated in TEME. This results in a file of average distances per constellation, i.e. set of TLEs. This file is read for its simulation in the circular model so that the average coordinate system center distance in an equivalent SGP4 simulation determines each satellite orbit's radius. Even though this leads to circular orbits that are more similar to the SGP4 ones, it has the disadvantage of requiring full simulation in another model, which is especially cumbersome for longer simulation durations. An outlook on improving this is given in the final chapter 6.

(2) To treat circular orbits as constant circles, they must be considered in an ECI coordinate system, as noted in section 2.3. As the TLE parameters describe the orbits in TEME, and `space_Veins` already has a coordinate transformation pipeline for the system, it was chosen as the coordinate system where positions are calculated. To calculate a satellite's position at a certain point of time using inclination, RAAN, circular anomaly and radius, the following mathematical approach was developed:

Instead of directly calculating satellite positions three-dimensionally in the TEME system, an approach was implemented to represent positions on circles in the orbital plane using only two parameters, which allows the calculation of positions from satellite movement as if it would happen in two dimensions. The two parameters are then transformed to 3D TEME coordinates based on the properties of the orbital plane.

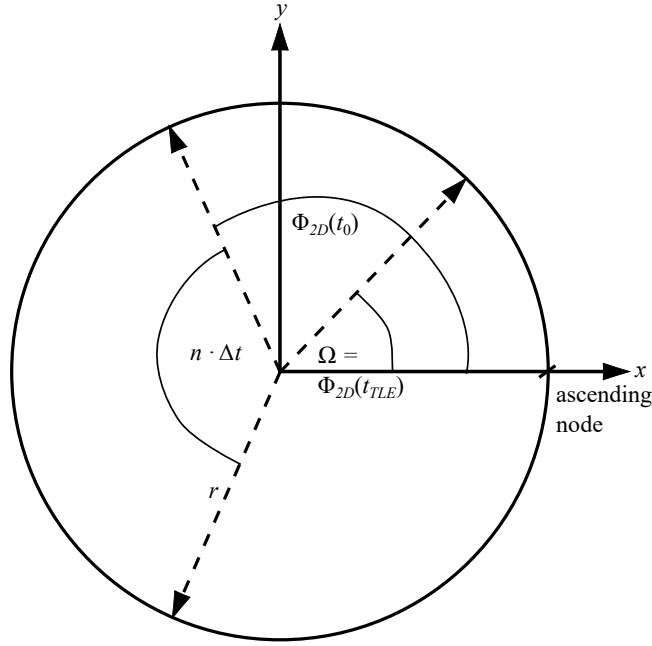
The implementation of movement as if it would happen on circular 2D orbits uses polar coordinates, i.e. radius and azimuth angle, and is depicted in figure 4.3. From the circular orbital elements (see figure 2.7) only requires radius  $r$ , mean motion  $n$  and the circular anomaly  $\Theta$  of the TLE epoch. The radius determines the size of the orbit and is constant, so the different positions on a fixed-size orbit are represented by different azimuth angles. Section 2.2 showed that velocity on the circle is uniform, thus the mean motion (converted from a TLE's *revolutions/day* to *rad/second*) is the actual angular speed of progress on the orbit. In 2D, this is the change of the azimuth angle  $\Phi_{2D}$ . Consequently, if the x-axis of the 2D coordinate system is oriented in the direction of the ascending node, which is achieved as described below, the azimuth angle at simulation start can be calculated from the azimuth angle at TLE epoch given by the circular anomaly (which is derived from a TLE by adding AOP  $\omega$  and true anomaly  $f$ ) as

$$\Phi_{2D}(t_0) = n \cdot (t_0 - t_{TLE}) + \Theta \mod 360^\circ, \quad (4.1)$$

and successively for every simulation step with step time  $\Delta t$  as

$$\Phi_{2D}(t) = n \cdot \Delta t + \Phi_{2D}(t - \Delta t) \mod 360^\circ, \quad (4.2)$$

or expressed w.r.t. the simulation start time as actually implemented:



**Figure 4.3** – Movement on a circular orbit in 2D using spherical coordinates: The size of the orbit is determined by the radius  $r$ . Due to the alignment of ascending node direction and x-axis, the position at TLE epoch is given by the circular anomaly  $\Theta$ , from which the azimuth angle at simulation start  $\Phi_{2D}(t_0)$  is calculated by adding  $n(t_0 - t_{TLE}) \bmod 360^\circ$  using mean motion  $n$ . The position of the next simulation step is specified by the azimuth angle  $\Phi_{2D}(t_0 + \Delta t) = \Phi_{2D}(t_0) + n \cdot \Delta t \bmod 360^\circ$ .

$$\Phi_{2D}(t) = n \cdot (t - t_{TLE}) + \Theta \bmod 360^\circ, \quad (4.3)$$

The uniform circular movement in 2D can be applied to 3D TME by firstly representing the orbital plane, and points in it with two parameters, using the parametric plane equation, which generally is [56, p. 118]:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} + \lambda \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} + \mu \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \quad (4.4)$$

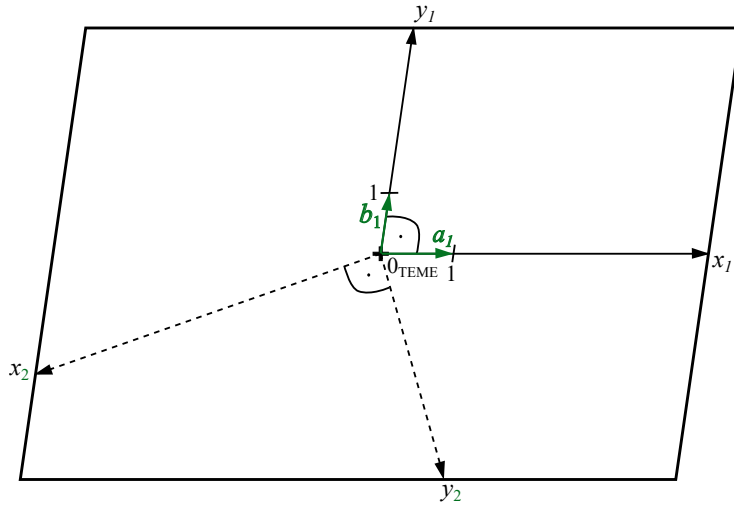
This achieves a description of the plane using a point and two vectors in it, if the vectors  $a$  and  $b$  are not parallel to each other - otherwise, only a line is represented. Because the coordinate system center always is in the orbital plane, as the intersection line of orbital plane and reference plane between ascending node and descending always passes through it, the equation can be simplified with  $x_1 = 0$ ,  $y_1 = 0$  and  $z_1 = 0$ , to:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \lambda \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} + \mu \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} \quad (4.5)$$

Thus, for known  $a$  and  $b$  the set of points in the plane is  $\{(\lambda, \mu | \lambda, \mu \in \mathbb{R})\}$ , i.e. the points in the orbital plane can be represented using two parameters. Under two conditions,  $a$  and  $b$  can serve as units of an x- and y-axis of a Cartesian 2D coordinate system in the orbital plane that can be imagined by shifting from a perspective that considers the three-dimensional TEME space to one that only considers the points in the plane, similar to how one can draw a valid Cartesian 2D system on a fully flat and unbendable piece of cardboard that remains a Cartesian 2d system regardless of its orientation in three-dimensional space; i.e. so that

$$a = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}_{TEME} \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}_{plane} \quad \text{and} \quad b = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix}_{TEME} \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}_{plane}, \quad (4.6)$$

and the position  $(1, 1)^T_{plane}$  is specified by  $(\lambda, \mu) = (1, 1)$ , i.e  $\lambda$  and  $\mu$  are the x and y coordinates of the imagined 2D coordinate system in the orbital plane. Firstly,  $a$  and  $b$  need to be perpendicular to each other, and secondly, they need to be unit vectors. Such imagined Cartesian 2D coordinate systems, and  $a$  and  $b$  are depicted



**Figure 4.4** – In any plane in 3D through the TEME coordinate center, an arbitrarily oriented 2D coordinate system w.r.t. the center can be imagined by extending two perpendicular vectors  $a$  and  $b$  from the center to an imagined x- and y-axis, on which unit vectors specify unit lengths - for example the one consisting of the axes  $x_1$  and  $y_1$  using unit vectors  $a_1$  and  $b_1$ , or the one consisting of the axes  $x_2$  and  $y_2$  as another example.

in figure 4.4. This combined with the transformation of polar coordinates to 2D cartesian coordinates presented in section 2.4 allows the use of the two-dimensional description of uniform circular movement with polar coordinates to determine  $\lambda$  and  $\mu$  for every simulation time step.

As stated above, any non-parallel  $a$  and  $b$  enable the two-parameter representation of the plane's points. However, if they are perpendicular but not normalized, every Cartesian 2D coordinate calculated from the two-dimensional movement description must be scaled to their magnitude when calculating the 3D position with equation 4.5. E.g. if the magnitude of  $a$  is two, any  $x$  value from the calculation as if in two dimensions needs to be halved to calculate the according  $\lambda$ . If the vectors are not perpendicular, it would be required to calculate how the units of the imagined Cartesian 2D coordinate system's axis can be composed from  $a$  and  $b$ , thus e.g.  $x_{2D} = 1$  would mean a combination of a certain  $\lambda$  and  $\mu$ . Hence,  $a$  and  $b$  being perpendicular unit vectors of the imagined Cartesian 2D coordinate system's axis leads to simpler transformation from the two-dimensional movement model to 3D coordinates.

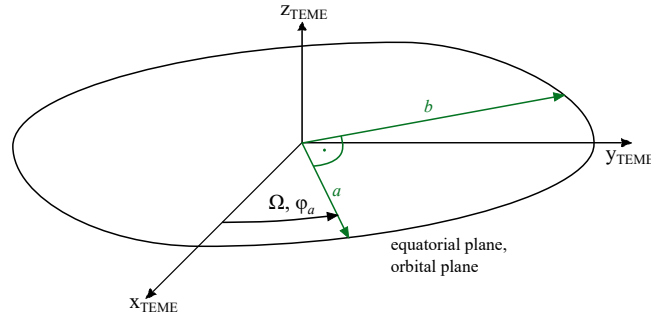
The condition of unit vectors can be fulfilled by normalizing any  $a$  and  $b$ . To achieve perpendicular vectors from the TEME system center, an approach using the circular orbital elements is applied:  $a$  and  $b$  are first specified with spherical (see section 2.4) instead of Cartesian TEME coordinates, and then transformed to Cartesian coordinates using equations 2.5 to 2.7. TEME's Cartesian x- and y-axis lie in the equatorial plane and x is the reference direction, the z-axis is Earth's polar axis [17]. Following the definitions of spherical coordinates w.r.t. the Cartesian axes, the unnormalized spherical vector  $a$  in the orbital plane is given by the RAAN  $\Omega$ . It is the counterclockwise angle to the x-axis in the equatorial plane, so has the same definition as the azimuth angle and thus for  $a$ , the azimuth angle  $\phi = \Omega$ , and the polar angle  $\rho = 90^\circ$ . Before normalizing, any vector  $a$  from the coordinate center in the direction specified by those angles can be used, e.g. with the magnitude of the circular orbit radius. I.e. with generally specifying a point in spherical coordinates as  $P_{spherical} = (\Phi, \rho, r)^T$ ,  $a_{spherical} = (\Omega, 90^\circ, r_{orbit})^T$ , where  $r_{orbit}$  refers to the radius of a concrete circular orbit and not the general  $r$  component of spherical coordinates.

Clearly, two vectors from the coordinate center that are in the orbital plane and perpendicular to  $a$  point in the direction of  $90^\circ$  additional degrees *in the orbit circle* clockwise and counterclockwise. For the ascending direction, i.e. the direction in which the satellite moves at the ascending node, three cases need to be considered when determining the vector perpendicular to  $a$ . (1) In the simplest case, the orbital plane and equatorial plane are equal. Then additional ascending  $90^\circ$  in the circle are  $90^\circ$  of azimuth angle - added or subtracted, depending on if the satellite moves clockwise, i.e. the inclination  $i = 180^\circ$ , or counterclockwise, i.e.  $i = 0^\circ$ , w.r.t the

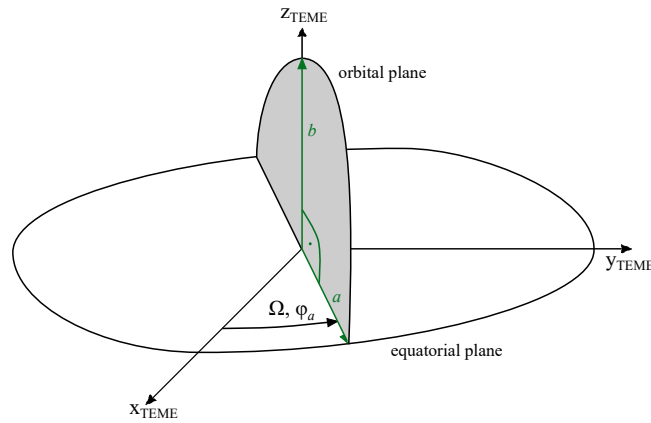
ascending node. This is depicted in figure 4.5. As the azimuth angle can only have values from 0 to  $< 360$  degrees, calculation needs to be done  $\text{mod } 360^\circ$ , i.e.

$$b_{\text{spherical}} = \begin{cases} (\Omega + 90^\circ \text{ mod } 360^\circ, 90^\circ - i, r_{\text{orbit}})^T & \text{if } i < 90^\circ, \\ (\Omega - 90^\circ \text{ mod } 360^\circ, |90^\circ - i|, r_{\text{orbit}})^T & \text{if } i > 90^\circ \end{cases} \quad (4.7)$$

When the orbit has an inclination of  $90^\circ$ , the half circle above the equatorial plane is perpendicular to it and crosses the z-axis. This is depicted in figure 4.6. Thus, the point with a perpendicular vector from the coordinate system center can be found at a polar angle  $\rho$  of  $0^\circ$ . In this case, any azimuth angle can be used, because the polar angle fully specifies the point to be on the z-axis in the positive direction, with the radius determining its exact position; which is also reflected by equation 2.7,



**Figure 4.5** – Unnormalized vectors  $a$  and  $b$  when equatorial and orbital plane coincide, for an inclination of  $0^\circ$ . The angles are RAAN  $\Omega$  and azimuth angle of  $a$   $\Phi_a$ .



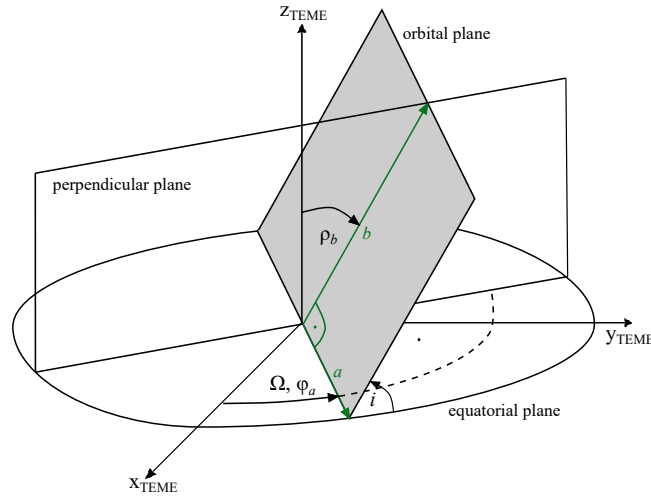
**Figure 4.6** – Unnormalized vectors  $a$  and  $b$  when the orbital plane is perpendicular to the equatorial plane, i.e. the inclination is  $90^\circ$  and  $b$ 's polar angle is  $0^\circ$ , while its azimuth angle can have any value.

which does not use the azimuth angle, and equations 2.5 and 2.6, which are always 0 with  $\rho = 0^\circ$ . Choosing  $0^\circ$  as fixed azimuth angle value,  $b_{spherical} = (0^\circ, 0^\circ, r_{orbit})^T$

For inclinations  $i \notin \{0^\circ, 90^\circ, 180^\circ\}$ , as depicted in figure 4.7, the general definition of inclination as an angle between two planes, a dihedral angle, can be applied, where *edge* means the plane intersection line: “the angle between two half-lines, one in each plane, drawn perpendicular to the edge from a common point” [20] (explained in section 2.2). Consequently for any inclination, any vector that is perpendicular to the intersection line in the orbital plane and thus to  $a$ , is also perpendicular to it in its horizontal, i.e. x and y, components. This applies to the desired  $b$ , which hence has an azimuth angle of  $\Phi_a + 90^\circ$  for counterclockwise movement w.r.t. the ascending node ( $i < 90^\circ$ ), and  $\Phi_a - 90^\circ$  for clockwise movement ( $i > 90^\circ$ ). Its polar angle  $\rho_b$  can be determined by applying the notion of the two half-lines again: The angle between the z-axis and the half-line in the equatorial plane is  $90^\circ$ , and the angle between both lines is  $i$ , thus the polar angle is  $|90^\circ - i|$ . Following this, the calculation of the spherical coordinates of unnormalized  $a$  and  $b$  can be summarized as:

$$a_{spherical} = (\Omega, 0^\circ, r)^T, \quad (4.8)$$

$$b_{spherical} = \begin{cases} (\Omega + 90^\circ \bmod 360^\circ, 90^\circ - i, r_{orbit})^T & \text{if } i < 90^\circ, \\ (\Omega - 90^\circ \bmod 360^\circ, |90^\circ - i|, r_{orbit})^T & \text{if } i > 90^\circ, \\ (0^\circ, 0^\circ, r_{orbit})^T & \text{if } i = 90^\circ, \end{cases} \quad (4.9)$$



**Figure 4.7** – Unnormalized vectors  $a$  and  $b$  when the inclination  $i \notin \{0^\circ, 90^\circ, 180^\circ\}$ . With a depicted inclination  $< 90^\circ$ ,  $b$ 's azimuth angle is  $\Phi_a + 90^\circ$ , and its polar angle  $\rho_b = 90^\circ - i$ .

which are transformed to Cartesian  $a$  and  $b$  via equations 2.5 to 2.7 and then normalized, so that the orbital plane is described by equation 4.5 using unit  $a$  and unit  $b$ . Any point in the plane can be described in the imagined 2D coordinate system with the two parameters  $\lambda \equiv x_{2D}$  and  $\mu \equiv y_{2D}$ , and transformed to TEME by solving equation 4.5 with those values.

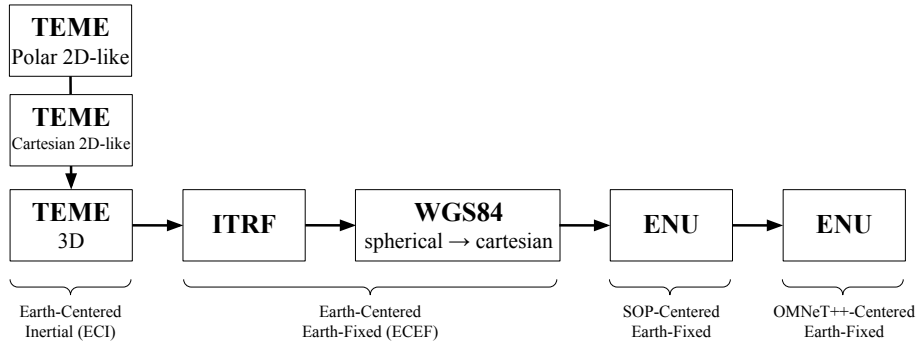
Lastly, the calculation of movement as if on a circular 2D orbit in polar coordinates needs to be integrated with the imagined Cartesian 2D system, which is done by applying equations 2.8 and 2.9 to equation 4.3, resulting in:

$$\lambda(t) = r \cdot \cos((t - t_{epoch}) \cdot n + \Theta) \quad (4.10)$$

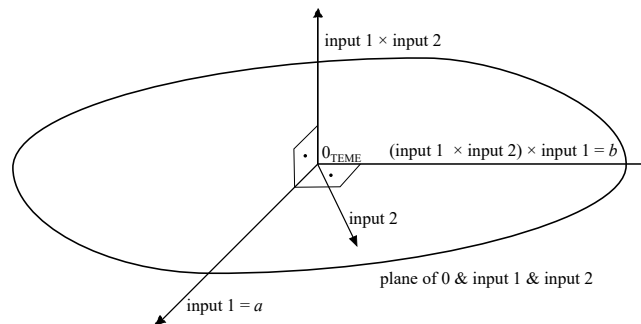
$$\mu(t) = r \cdot \sin((t - t_{epoch}) \cdot n + \Theta), \quad (4.11)$$

which are inserted in equation 4.5 to get the Cartesian 3D TEME coordinate of time  $t$ . From there, the same coordinate transformation pipeline used for SGP4 TEME coordinates is applied. The whole transformation pipeline from polar coordinates in the orbital plane to ENU coordinates w.r.t. the OMNeT++ coordinate system center is shown in figure 4.8.

Similar to the Keplerian orbit model, this was implemented by adapting the SGP4Mobility class to the CircularMobility class. Like the satellites based on the SGP4Mobility, each satellite is initialized from a TLE, which is managed by the SatelliteInserter class. As for SGP4, it reads a file with a list of TLEs per simulation configuration and run, which is specified using the same OMNeT++ parameter pathToTleFile used for SGP4. The according functions are adapted to initialize satellite modules of a TLE file based on the CircularMobility with the average radii from previous SGP4 simulation, which are given in a CSV file specified by an additional OMNeT++ parameter called avgSGP4RadiiPath for the



**Figure 4.8** – Transformations applied to coordinates of circular orbits; fully in C++



**Figure 4.9** – Initialization of a `CirclePlane` instance as implemented because of a previous approach that used two SGP4-derived points on the orbit that is to be modeled: unnormalized perpendicular  $a$  and  $b$  from their vectors from the coordinate system center in the orbital plane.  $b$  is calculated by applying the cross-product twice.

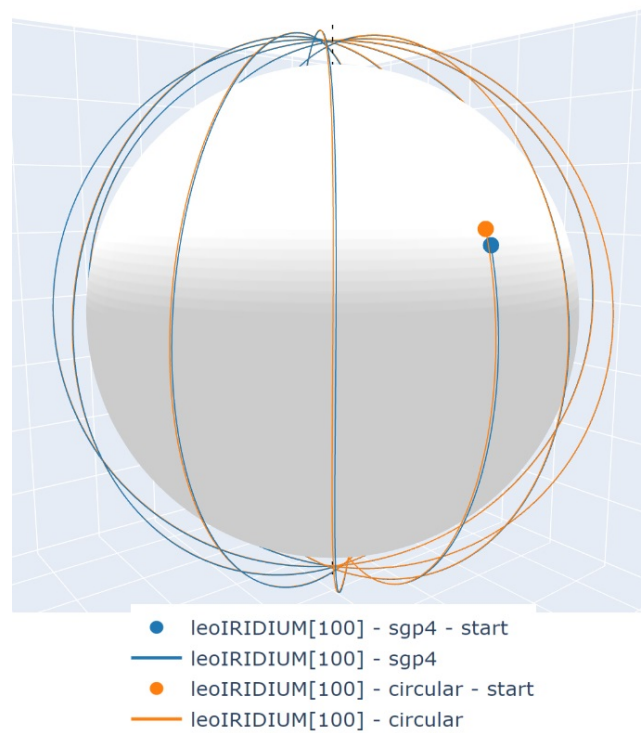
`SatelliteInserter`. For handling the calculations of coordinates in 2D and their transformation to Cartesian 3D TEME, a separate `CirclePlane` class was implemented, which, as an artefact from a previous implementation, can be initialized using any two points in the orbital plane with vectors from the coordinate system center that are not parallel to each other, from which unit  $a$  and  $b$  are calculated as depicted in figure 4.9. The vector of the first point is directly used as unnormalized  $a$ . (1) Using the cross-product of it and the vector of the second input point, a perpendicular vector to both is calculated, i.e. one perpendicular to the orbital plane. (2) On this vector and the unnormalized  $a$ , the cross-product is applied again, yielding the unnormalized perpendicular  $b$  vector to both which is in the orbital plane again. By using the vector calculated in (1) as first component of the cross-product, the right-hand rule ensures that is on the same side of  $a$  as the second input vector. If the components of the second cross-product would be changed, the resulting vector would point in the opposite direction. However, both input points/vectors are already calculated as previously described using equations 4.5 to 4.11 in the `CircularMobility`, so the initialization of the `CirclePlane` just recreates  $b$  which was already given by the second input point. Finally,  $a$  and  $b$  are normalized. The previous implementation approach used two points of the orbit which were calculated via SGP4. The resulting orbital plane thus was only indirectly related to inclination and RAAN. In contrast, in the current implementation the orientation of the orbital plane is directly derived from those parameters and no use of SGP4 beyond radius determination is required.

The calculations of the satellite position on the orbit in the imagined 2D coordinate system at a certain point in time and its transformation to 3D are implemented exactly as they were described above, using the other `CirclePlane` initialization parameters radius, mean motion converted to *rad/second* from a



TLE's *revolutions/day*, and the azimuth angle  $\Phi_{2D}$  at the simulation start time, which is calculated from a TLE's true anomaly, AOP, epoch and mean motion. For each simulation time step, the `CirclePlane` of a `CircularMobility` instance is called to get a TEME position, while the rest of the class remains the same as in the `SGP4Mobility`. The `SatelliteInserter`'s `mobilityType` parameter needs to be set to "Circular", and the satellite module's mobility submodule's `typename` parameter to the "CircularMobility" class.

This approach was validated by (1) calculating the resulting distances to the coordinate system center in TEME. In theory, for a circle they all should be equal. However, this validation showed that they differ by a few millimeters. As the distances were as big as around 6600 to 7700 km, and were calculated by spherical-Cartesian and 2D-3D conversions, this can be attributed to floating-point-based calculations. The differences are so insignificant relative to the distances, that the orbits of the model can be considered sufficiently circular. (2) Secondly, as depicted in 4.10, they again are very similar to at least nearly circular orbits with very low eccentricity on a macro scale, which shows that the TLE parameters were most likely correctly adapted to the circular model. These position differences will be examined in detail in chapter 5.



**Figure 4.10** – Visualization in the ECEF system ITRF, of the orbits of the IRIDIUM-100 satellite, with an eccentricity of ca. 0.0002, in the SGP4 and Circular model, simulated for 12 hours. Even though at the same start time, the position is clearly different, the trajectories from there on are so similar on this scale that they seem to overlap, e.g. on the very left. In contrast to figure 4.2, the parts of the trajectories directly in the middle are visibly different. Because in this instance, the simulation start wall-clock time is at a quite high distance to the epoch, there is a significant difference between the starting positions.

---

## Chapter 5

# Evaluation

---

To assess the effects of exchanging SGP4 with the Keplerian or circular orbit model in `space_Veins` on communication metrics, and link them to the caused differences of satellite positions, a simulation study involving three surface locations at different latitudes and five different LEO constellations was conducted. More concretely, in the unifying ENU/OMNeT++ coordinate system, it studied positional differences via Euclidian distance, and differences of communication by assessing how availability periods and the propagation delays in them change. Using the 20 km maximum distance of SGP4 positions to real satellite orbits in one-day propagation from a TLEs epoch estimated from the analyzed related work, this not only allows to state how positions differ from SGP4, but also to approximate how they differ from the represented real orbits. ENU coordinates were used because (1) in contrast to an ECI perspective for which SGP4 and the circular model use TEME, but the Keplerian model uses GCRS, orbits in all models result in ENU coordinates in `space_Veins`' approach, and (2) the communication properties are directly related to positions in ENU, not coordinates in the intermediate coordinate systems.

Because the focus of the study were differences between the satellite mobility models, ground mobility was not considered, i.e. communication to static ground stations was simulated. For the same reason, the study was limited to one ground station per surface location instead of additionally considering multiple ground stations in its proximity. Without the need to evaluate multiple ground-satellite links per orbit and availability period w.r.t. a location, the study could purely be focused on studying a vast set of orbits with different parameters.

To also accurately assess the runtime of the models, i.e. for SGP4 and the circular orbit model the simulation in OMNeT++, and for the Keplerian model trace-precomputation in Python and OMNeT++ simulation combined, all metrics of this evaluation were calculated after the simulation via Python, instead of including their calculation in the simulation. As all metrics can be calculated from ENU coordinate

data, no ground-satellite communication was simulated; even though this is a feature of `space_Veins`. I.e. the communication metrics calculations from the simulation results assess which visibilities occurred in the simulation, but only if communication *would have been* possible, and which propagation delays *would have* characterized the communication.

In the following, firstly the simulation parameters are presented, i.e. considered satellite constellations, surface locations and parameters common to all configurations, then the implementation of configurations via embedding in a workflow defined with the Snakemake [57] tool is described. This setup of the simulation study is followed by the definition of the metrics which were calculated from its data, for which lastly the results are presented and discussed. It is important to note that the thesis originally intended to compare the different constellations and even subconstellations of them in detail, however did its time frame only allow this in-depth initially for the comparison of positions between SGP4 and the alternative models. Hence, after the section on them, medians and averages were used widely, while only exceptional results of constellations/subconstellations combined with certain locations are discussed further.

## 5.1 Satellite constellations

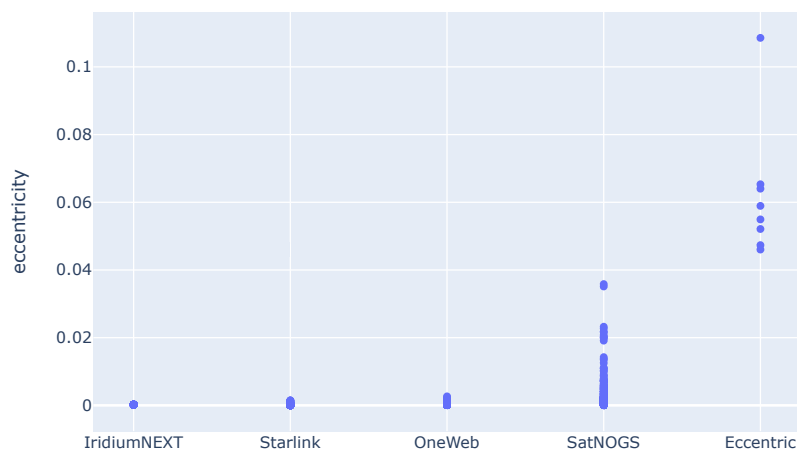
Five different constellations, i.e. sets of TLEs, were compiled from the LEO TLEs available from NASA/NORAD at noon on the 23rd October 2023. Firstly, they can be grouped by their eccentricity: Four constellations consist of nearly circular orbits, which following Van Eylen et. al [58] are orbits with an eccentricity of 0.04 and below. This applies to the full sets of TLEs that were available for the *IridiumNEXT* (80 TLEs) and *OneWeb* (635 TLEs) constellation. The *Satellite Networked Open Ground Station (SatNOGS)* and *Starlink* constellations contained a small number of TLEs with an eccentricity larger than 0.04, which were removed from the TLE sets for the simulation study, leading to a set size of *Starlink* of 4906. Additionally, the *SatNOGS* TLEs were parsed with the Python `sgp4`<sup>9</sup> library to evaluate their *alta* parameter, i.e. their highest altitude occurring at the apogee, to remove all TLEs representing orbits that surpass an altitude of 2000 km and thus cannot be classified as satellites on a purely Low-Earth Orbit according to Qu et. al [59], leading to a TLE set size of 908. As is implied by its name, *SatNOGS* is not an actual constellation of satellites related by design, but a collection of TLEs from open source ground stations tracking the contained satellite orbits [60]. This shows in the remainder of this section, where *SatNOGS* has the most varied characteristics among the nearly circular constellations. The *SatNOGS* TLE set was simulated together and thus

<sup>9</sup><https://github.com/brandon-rhodes/python-sgp4>

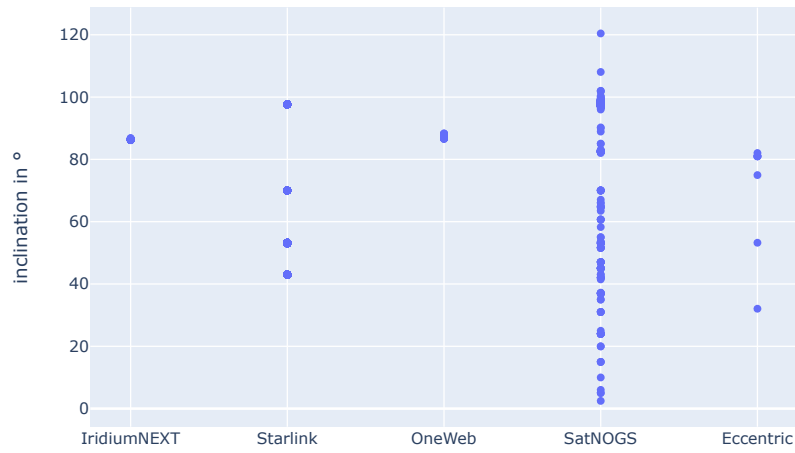
considered a constellation in the simulation study, to evaluate if effects of the orbit models can be stated with similar clarity for a set of such diverse orbits, or if the behavior of their concurrent simulation is less predictable than for the other, actual, constellations. From the in total 8 TLEs with an eccentricity larger than 0.04 from the original *Starlink* and *SatNOGS* TLE sets, the fifth constellation, which was named *Eccentric*, was created, with eccentricities ranging from ca. 0.046 to ca. 0.109. The reason for this separation is the circular orbit model, because it is likely to perform worse in approximating orbits that are more elliptical, i.e. have a higher eccentricity.

Firstly considering the eccentricities in detail, which are depicted in figure 5.1, it is notable that the orbits of the other nearly circular constellations have rather similar eccentricities close to 0, whereas the *SatNOGS* orbits have various eccentricities up to above 0.02. *SatNOGS* even includes one orbit with an eccentricity that comparatively is just below the defined limit of 0.04, with a value closer to most of the *Eccentric* eccentricities than to those of *IridiumNEXT*, *Starlink* and *OneWeb*.

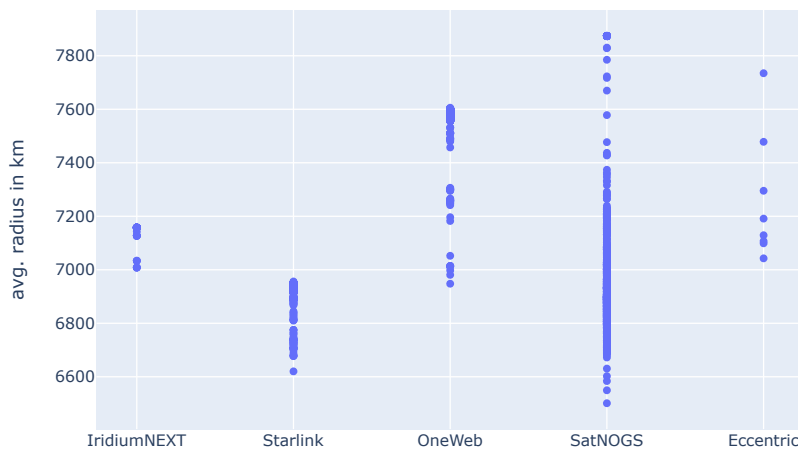
The nearly circular constellations can further be characterized by their inclinations, which are depicted in Figure 5.2. This data shows that the *Starlink* constellation consists of four inclination groups, whereas *IridiumNEXT* and *OneWeb* inclinations are very similar, and the *SatNOGS* inclinations are spread out across the value range. To study if certain positional or communication differences can be linked to inclination differences, four *Starlink* subconstellations were created by splitting the TLE set into the inclination groups *StarlinkLowest*, *StarlinkLow*, *StarlinkHigh* and *StarlinkHighest*, where the suffixes refer to the magnitude of the inclination ranges. The concrete ranges are presented in table 5.1 at the end of this section that presents all considered properties of the constellations.



**Figure 5.1** – Eccentricities of the 5 constellations considered in the simulation study



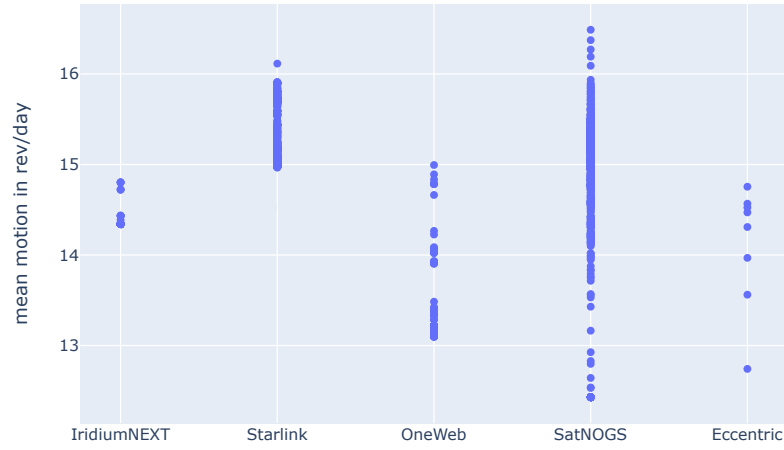
**Figure 5.2** – Inclinations of the 5 constellations considered in the simulation study



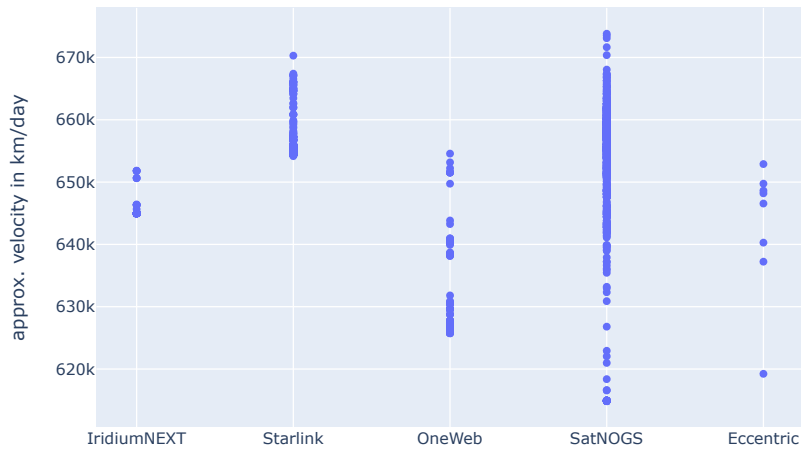
**Figure 5.3** – Average SGP4-TIME radii of the 5 constellations considered in the simulation study over a 12-hour period

The nearly circular constellations can also be characterized by the average radii of their orbits calculated from 12-hour SGP4 simulation in TME as depicted in Figure 5.3. This data shows two radius groups in the *IridiumNEXT* constellation and three radius groups in the *OneWeb* constellation. *Starlink* and *SatNOGS* TLEs are spread out across the value range. With equation 2.1 and the approximation of the semi-major axis length  $a$  as  $a = 2r$  for these nearly circular orbits, orbits with higher average radii have a lower velocity. One hypothesis for the simulation

study was that with higher velocities, more movement happens in a simulation time step, and hence higher positional differences occur. To research this, two *IridiumNEXT* subconstellations *IridiumNEXTlow* and *IridiumNEXTHigh*, and three *OneWeb* subconstellations *OneWebHigh*, *OneWebMiddle* and *OneWebLow* were created, where the suffix refers to the magnitude of their average radius ranges in TEME.



(a) mean motion



(b) circularly approximated velocity

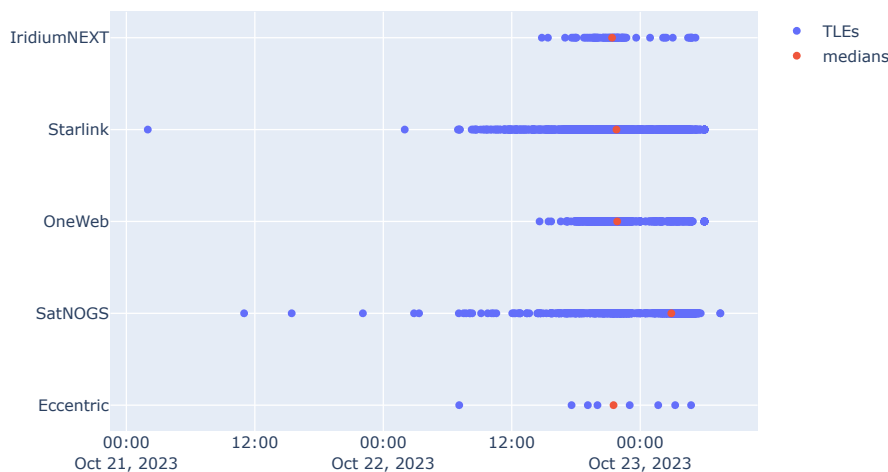
**Figure 5.4** – Mean motion and approximate velocity of the 5 constellations considered in the simulation study

$$v_{circ} = u \cdot n = 2\pi r \cdot n \quad (5.1)$$

The calculation of the approximate velocity in km/d on a circular orbit, from mean motion  $n$  and average radius  $r$  via equation 5.1 that uses the circumference  $u$ , shows that mean motion and average radius are linked and thus the above approximation is valid, as depicted in figure 5.4 (a) and (b): The same groups as for the average radii also appear in both plots, however in reverse order as in figure 5.3, as a higher radius means lower velocity. As the *Eccentric* constellation only consists of 8 TLEs, no grouping was attempted.

From the Keplerian orbital elements, also AOP and RAAN could be used to characterize the orbits of the constellations. However, among the TLEs of the subconstellations established above, generally a variety of their values occurs. As the amount of constellations and subconstellations already leads to quite extensive comparative evaluation, the decision was made to not form sub-subconstellations based on AOP and RAAN, which would have further multiplied the evaluation efforts. Consequently, linking of these TLE parameters to characteristics of the positional and communication metrics was generally not attempted. Just for the RAAN, some insights will be presented in the results of the positional differences of the Keplerian and circular model to SGP4 and the *OneWebHigh* and *IridiumNEXThigh* subconstellations.

The TLEs of the constellations have various epochs, as depicted in figure 5.5. The related works clearly show that the accuracy of SGP4 simulation decreases with increasing time distance to them, hence the simulation period should be close to



**Figure 5.5** – TLE epochs and epoch median for the 5 constellations. Three outliers have been removed for this visualization to achieve a greater resolution: One *Starlink* epoch on the 10th October, and two *SatNOGS* epochs on the 4th and 17th October.



the epochs. Adhering to this, it was decided to relate the start time of a simulation, which applies to all TLEs of a constellation, to their median, which also is depicted in the figure. It was calculated for the whole constellation and also applied to the subconstellations, so that they also can be considered together as if they were simulated together. Other TLE parameters only specify the orbits in SGP4 and require understanding of the mathematical intricacies of the model, which is beyond the scope of this thesis. Consequently they were not considered.

Because of the number of 4906 TLEs in the *Starlink* constellation, which is a very large number of satellites to simulate, the inclination groups were used to reduce the TLE set size. Each *Starlink* subconstellation was reduced by removing about 4/5 of the TLEs from each, via random selection. Not every set was dividable by 5, hence 1-3 additional TLEs were removed per subconstellation. As the *Starlink* TLEs only form clear groups w.r.t. inclination and not the other considered parameters, this can reduce the set size considerably while maintaining a variety of orbital characteristics in the constellation.

The constellation characteristics described above are summarized in Table 5.1, which also gives concrete value ranges. To give the radius average per orbit a more intuitive meaning, it was transformed into the average altitudes over the surface, using the 6 371 008.7714 m "mean radius of the three semiaxes" specified for the WGS84 ellipsoid [61].

Constellation	Median epoch on 22.10.23	Subconstellation	#TLE	$i$	Avg. (per orbit) alt. (km)
<i>IridiumNEXT</i> (subconst. via alt.)	21:21:57	IridiumNEXTlow	7	86.156°-86.676°	637-663
		IridiumNEXThigh	73	86.391°-86.859°	755-787
<i>Starlink</i> (subconst. via $i$ )	21:46:37	StarlinkLowest	228	42.998°-43.007°	306-568
		StarlinkLow	625	52.997°-53.243°	249-570
		StarlinkHigh	81	69.997°-70.004°	362-585
		StarlinkHighest	46	97.652°-97.659°	571-574
<i>OneWeb</i> (subconst. via alt.)	21:50:54	OneWebLow	9	86.538°-87.997°	577-681
		OneWebMiddle	27	86.915°-87.541°	811-935
		OneWebHigh	599	87.840°-88.321°	1086-1235
<i>SatNOGS</i>	+ 1 day, 02:54:16	-	908	2.468°-88.9181° & 90.043°-120.425°	130-1477
<i>Eccentric</i>	21:30:18	-	8	32.055°-82.041°	672-1364

**Table 5.1** – Constellations, subconstellations, and their parameters median epoch, number of TLEs, inclination  $i$  (rounded) range and range of altitudes averaged per orbit (rounded), which is derived from the orbits' average radii in TEME in a 12-hour SGP4 simulation and the WGS84 ellipsoid's mean radius of the three semiaxes. For SatNOGS, two inclination intervals are specified to clearly show how polar some of its orbits are.

## 5.2 Surface locations

As presented in the fundamentals in chapter 2, a major difference between the orbit models is that SGP4 considers Earth's non-uniform gravitation caused by its oblateness, whereas the reason for satellite movement in the Keplerian and circular model, as special case of it, is the gravitation from Earth as a point-like mass in the coordinate system center. To study if this affects the between-model differences of the positions calculated w.r.t. a surface location to achieve ENU coordinates, the simulation study evaluated three different surface locations at different latitudes. For their specification, WGS84 latitude and longitude are used.

Although clearly, the same position in an ECI coordinate system is transformed to different ENU positions relative to different surface locations, the positional differences caused by the models are already present in the ECI systems. Theoretically, they are maintained throughout the transformation pipeline to relative coordinates and thus can be compared for different surface locations. As is presented later in this chapter, the question if the positional differences between the models are actually maintained through coordinate transformations in practice was also evaluated in the simulation study. In contrast, between-model differences of communication availability periods and propagation delays therein cannot directly be compared for different surface locations, as even in the same orbit model, different availability periods occur for the different locations. However, relative metrics on their differences between the models can still be considered.

Because oblateness means that Earth has different radii at different latitudes, the three locations were selected to represent the full latitude range that is supported by the UTM projection [62] required to realize a planar road network in SUMO: firstly, an equatorial latitude of  $0^\circ$ . To select a location where an actual complex road traffic network might be simulated, the position  $0.000127^\circ\text{N}$ ,  $78.449996^\circ\text{W}$  was selected, which is situated in Ecuador's capital Quito. Compared to the equatorial position in the proposal of this thesis, it was slightly adapted by a few meters of ground distance, to actually be on a road. The same applies to the selected position in the next paragraph.

Secondly, a latitude near the middle of the hemisphere, i.e.  $45^\circ$  was chosen. Another consideration for this was that, as described by Gill and Montenbruck [63, p. 32], a satellite orbit with inclination  $i$  "covers geographical latitudes between a minimum of  $\Phi = -i$  and a maximum of  $\Phi = i$ ". Because "an inclination of more than  $90^\circ$  means that the satellite's motion is retrograde, its direction of revolution around the Earth being opposite to that of the Earth's rotation." [63, p. 25], locations at a latitude  $\pm\Phi$  can only theoretically be directly flown over by orbits with an inclination in the interval  $[\Phi, 180^\circ - \Phi]$ . To theoretically enable communication of the satellites of most constellations with a location at a latitude near  $45^\circ$ , the latitude was selected

according to *Starlink*'s lower inclination limit of  $42.998^\circ$  (different from the  $42.990^\circ$  in the proposal of this thesis, which can be attributed to the use of more recent TLEs with different parameters). The lowest limit, which is given by the *SatNOGS* constellation at  $2.468^\circ$ , would not allow choosing a latitude near  $45^\circ$ , and thus it was accepted that satellites on some orbits of the constellation cannot communicate with the second location. Hence, the position  $42.698156^\circ\text{N}$ ,  $23.319892^\circ\text{E}$  in Bulgaria's capital Sofia was selected.

Lastly, a location near one of the poles at  $90^\circ$  was to be chosen. To select a location where any traffic might occur, antarctic research stations were considered, from which Kunlun Station was selected in the proposal of this thesis, at  $80.417222^\circ\text{S}$ ,  $77.116111^\circ\text{E}$ . However, because the application of UTM is limited to  $80^\circ$  in the southern hemisphere [62], this had to be adapted to  $80^\circ\text{S}$ ,  $77.116111^\circ\text{E}$ , which is roughly 46 km north of the research station. Nevertheless, this allows the study of the models' differences in polar regions. The latitude of  $80^\circ\text{S}$  limits the number of satellites that can theoretically communicate with the location further, completely excluding the *StarlinkLowest*, *StarlinkLow* and *StarlinkHigh* inclination subconstellations.

For simulating only communication to one static ground station at the surface locations, no road traffic network was required. To prevent potential issues with SUMO when the traffic network files only contain one node and no edges, a second node and an edge to it were added, which do not play a role in the actual simulation. In the `.nod.xml` file, the first node representing the static ground station has the coordinates as specified above, where southern latitudes and western longitudes translate to negative values. The interpretation of these coordinates by SUMO/OMNeT++ always results in an OMNeT++ coordinate of  $(x, y, z) = (25\text{ m}, 25\text{ m}, 0\text{ m})$ , which coincides with the configured SOP position w.r.t. which ENU coordinates are calculated. Thus, no distance error to the represented ground station is introduced, as would happen for other OMNeT++ positions on the East-North plane tangential to the WGS84 ellipsoid. For the auxiliary node, positions in the South-East of the respective surface locations were configured, to ensure that the second node always is below and to the right of the first one in the OMNeT++ playground. This is required so that the first node always has the same position in the playground, and is not shifted when the second node unexpectedly is above the first one – which did happen when the auxiliary node was just placed in the East of the first one, because of the UTM projection applied to the WGS84 coordinates. E.g. for Quito, the content of the `.nod.xml` is:

---

```

1 <nodes>
2   <node id="n0_0" x="-78.449996" y="0.000127" type="priority"/>
3   <node id="n1_0" x="-78.2" y="0" type="priority"/>
4 </nodes>

```

---

Of the other traffic network files, only the `.net.xml` needs to be adapted to the different `.nod.xml` file via the `createRoadNetwork.sh` script.

The locations in Quito and Sofia also served as indicators for deciding which minimum elevation angle should be used to evaluate communication availability. On one hand, in cities line of sight obstructions by buildings are likely, so it clearly should be larger than  $0^\circ$ . On the other hand limits a high minimum elevation angle the area of visibility over a surface location severely, and thus limits the amount of visibility periods that could be evaluated in the simulation study. For deciding on a minimum elevation angle approximately representative of various in-city road positions near the surface locations of the simulation study, not just the locations themselves, the information given for the cityscape at the locations in Sofia and Quito by Google Street View was considered, which is depicted in Figures 5.6 and 5.7. Because of the question of balancing the static minimum elevation angle, it was decided to not adapt it to the banking building in Sofia as the highest visible one. Instead, it was adapted to the 4-story building marked by a yellow box on the figure, for which many buildings of similar heights and road locations next to them exist down the road. It is higher than all buildings visible in Quito. Estimating its height to 12 m, and estimating the ground distance from the middle of the road half next to it to the start of the building to 6 m, solving equation 5.2 for the elevation angle  $\alpha$  results in around  $63.43^\circ$ .

$$\tan(\alpha) = \frac{\text{opposite}}{\text{adjacent}} = \frac{\text{obstruction height}}{\text{ground distance}} \quad (5.2)$$

As the road and the sidewalk are narrower in Quito, the 2-to-1 height-to-ground-distance ratio estimated above may apply to the highest visible building, but it may also be slightly higher than that. However, the views in the road directions are much less obstructed at both locations. Clearly, one elevation angle cannot accurately model such scenarios. To achieve an area of visibility above the surface location that is similar to the reality in Sofia, the unrounded calculated elevation angle was simply halved to  $31.73^\circ$ , because other obstruction estimates can be hardly done from the figures. It is important to note that because the communication metrics are calculated after and not during the simulation, this is not a configuration of the minimum elevation angle implemented in `space_Veins`, but a configuration for the metrics calculation.



(a)



(b)

**Figure 5.6** – Obstructions at  $42.698156^{\circ}\text{N}$ ,  $23.319892^{\circ}\text{E}$  in Sofia from Google StreetView. The minimum elevation angle of  $63.43^{\circ}$  was estimated for obstructions like the building in the yellow box and ground positions like in the middle of the road half next to it.



(a)



(b)

**Figure 5.7** – Obstructions at 0.000127°N, 78.449996°W in Quito from Google StreetView.



### 5.3 General simulation parameters

After a preliminary assessment of the maximum resource usage of the simulations and calculation of metrics, it was decided to simulate the constellations for 12 hours. Adding a simulation warm-up time of 15 s, and cool-down time of 5 s, the total simulation time is 43 220 s. The simulation steps were configured to 1 s.

Because the related works showed that the accuracy of SGP4 decreases with increasing time distance to the TLE epoch, and the preliminary simulations showed that the positional differences of the Keplerian and circular model to SGP4 increase in the same way (presented in detail in the results section 5.6), it was decided to place the median epoch time of the constellations in the middle of the simulation period. This means that for each constellation, the simulation wall-clock start time is  $(43220/2) - 5 = 21605$  s before the median epoch of its TLEs, i.e. 6 h minus 5 s, where the 5 s are subtracted to balance out the difference between warm-up and cool-down time. This is specified as suffix of the TLE set .txt files in the *YY-MM-DD-HH-MM-SS*, i.e. *Year-Month-Day-Hour-Minute-Second*, format, so that the file contents can remain pure TLE sets.

Lastly, it was required to configure in which coordinate systems positions need to be recorded by the OMNeT++ simulation. Firstly, to derive the radii of the circular orbit model from SGP4, TEME coordinates are required. Secondly, to evaluate if positional differences between the models are maintained throughout coordinate transformations, the transformation steps from ITRF to ENU were chosen, as they are common to all the orbit models. Because of this, ITRF coordinates are required, in addition to ENU coordinates, which are required for this and to calculate between-model differences of metrics of communication and the positional differences that are the direct cause of them.

### 5.4 Simulation configuration via Snakemake

*Snakemake* [57] is a workflow management tool that allows the implementation of data processing pipelines in a Python-based language, which can be executed in automated, parallel and reproducible fashion. Fundamentally, it consists of *rules*, which process certain input files into certain output files via Python code in scripts or directly in the Snakemake language, R scripts, or shell commands. By building a Directed Acyclic Graph (DAG), Snakemake automatically identifies which rules need to be executed to produce the input files for other rules if they don't exist. To fully ensure the reproducibility of the simulation study, it also is implemented in a *Singularity* [64] container, which together with some rules that complete its initialization provides all required dependencies. In addition to one or multiple



files containing the rules, Snakemake also uses a configuration file where values to be used in the workflow's execution can be statically specified. Not only the configuration of the simulation study was implemented via Snakemake, but also its execution and the calculation and plotting of positional and communication metrics and their between-model differences via Python scripts. Because this only automates the parametrization and execution of the required shell commands, which may also be executed manually, it is not explained in full detail in this thesis. The implemented processing pipeline can be summarized as (1) generation of simulation configuration, (2) ITRF trace generation for the Keplerian model, (3) simulation execution, and (4) calculation of metrics and plotting them. As the first step completes the definition of how simulations were executed, for it, further details are provided here.

As detailed in section 2.1, the simulation configuration consists of the SUMO traffic network files and the `omnetpp.ini` file containing configurations of the OMNeT++ modules. To realize a simulation study with the parameters in the previous sections, three configurations of these files were implemented, i.e. one for each surface location, of which only one exists at a time and is adapted when changing the current surface location. The current location is configured in the Snakemake configuration file. For the two-node traffic network described in the previous section, this means using three variants of the `.nod.xml` and adapting the `.net.xml` to it. This was realized by creating a `.nod.xml.template` file with a placeholder for the node specifications, which all three are statically specified in the Snakemake configuration file. When the location in the configuration file is changed, a Snakemake rule is executed that creates an updated `.nod.xml` from the template, after which a Snakemake rule that updates the `.net.xml` via the `createRoadNetwork.sh` script is run. The `.nod.xml.template` file is required because Snakemake does not allow cyclicity, i.e. an old `.nod.xml` cannot be the input to create an updated `.nod.xml`.

The other component of the simulation configuration, the `omnetpp.ini`, is created using a Snakemake rule that executes a Python script with the same approach. I.e. it uses an `omnetpp.ini.template` file which contains the "General" configuration of the simulations, with the module configurations that apply to the all simulation instances, and a "Debug" configuration used to create debug variants of all constellation-model configurations described below. Theoretically, the same `omnetpp.ini` can be used for all the locations, as the placement of the traffic network is fully configured by the files described above. However, a parameter using the location names was added to the SOP module to identify the simulation results with the locations. As it is the same for all simulations w.r.t. the location, it is a part of the "General" section already contained in the template, which is overwritten by the updating Snakemake rule / Python script. But this is only a minor part of the creation of the `omnetpp.ini`.

The remaining part of the `omnetpp.ini` is automatically created, containing the configurations for simulating the total number of 11 constellations/subconstellations using the 3 mobility models, where the parent constellation is not simulated when subconstellations exist, but their results are combined for metric evaluation. Thus, using a OMNeT++ simulation configuration template file, 33 configurations for full simulation are generated, and 33 that extend the generally applicable debug configuration - resulting in configurations named `("Debug-")constellation-mobility_model`. This was implemented so that in the Snakemake configuration file, only the constellations need to be specified, while the updating script handles creating configurations for them. This includes: For the `SatelliteInserter` module, setting the `mobilityType` parameter specifying the used mobility model, as to determine how satellites are created, and setting the `pathToTLEfile` parameter for the SGP4 and circular orbit model - which also requires the `avgSGP4RadiiPath` - or alternatively the `pathToTracesDir` parameter for the Keplerian model; as well as specifying the simulation start time with the `wall_clock_sim_start_time_utc` parameter. For the satellite module, the used mobility model needs to be specified as well with the `mobility.typeName` parameter. Because the mobility models do not include any randomness, each constellation - mobility model configuration corresponds to one simulation run.

## 5.5 Metrics

To assess the **positional differences** of the Keplerian and circular model to SGP4, the Euclidian distances between the coordinates of each simulation time step and orbit of the alternative models to SGP4 were calculated. This was done in ENU/OMNeT++ coordinates, but also in ITRF to evaluate if and how the distances change through the ITRF-to-ENU/OMNeT++ coordinate transformations. Only the transformations from ITRF are considered instead of the initial ECI coordinate systems of the models, because ITRF is used for all models, while in ECI for SGP4 and the Circular model, TEME is used, but the Keplerian model uses GCRS. As will be further elaborated in the next section, to compare the magnitude of to-SGP4 distances in backward and forward propagation from the TLE epoch, (1) fitting of linear functions to the evolution of distances in both segments was attempted, where from the general equation  $y = a * bx$  mainly  $b$  was considered to characterize the evolution, and is called *growth factor*. (2) As the to-SGP4 distances display considerable periodicity, the distances in periods of 6 hours of forward and backward propagation were averaged per orbit which provided such a forward or backward period, based on the time of its epoch w.r.t. the simulation start, of which averages for whole constellations and subconstellations were formed.

Because the growth factors proved to be prone to inaccuracies, finally worst-case estimates of the distances in around 6 hours of backward and forward propagation per subconstellation, or constellation where there are no subconstellations, were calculated. For this, the highest distance to SGP4 that occurred in a subconstellation/constellation in the backward/forward periods was identified. For orbits where the magnitudes of to-SGP4 distances are not dominantly characterized by periodicity but also growth that is presumably linear has a high influence, these estimates were linearly extrapolated to 24 hours. The resulting values were then combined with the 20 km maximum estimate of SGP4 distances to real satellite orbits in one-day propagation from the related works in another worst-case approach, which assumes that the to-SGP4 and SGP4-to-real-orbit distances simply add up, to provide a worst-case estimate of the Keplerian and circular distances to real satellite orbits in one-day backward/forward propagation.

The fundamental **metric w.r.t. communication** calculated for each orbit is the *elevation angle*  $\Phi$  that results from the satellite position w.r.t. each evaluated surface location. For this, formula 5.3 [56, adapted from p. 62] to calculate the angle between two vectors was used, which requires the vector from the surface location at  $(x, y, z) = (0.025 \text{ km}, 0.025 \text{ km}, 0 \text{ km})$  in ENU/OMNeT++ coordinates to the satellite position recorded in km called  $\vec{s\hat{a}t}$ , and its projection to the East-North plane by setting its z coordinate to 0,  $\vec{s\hat{a}t}_{EN}$ , where the dot product is used in the numerator.

$$\theta = \arccos \left( \frac{\vec{s\hat{a}t} \cdot \vec{s\hat{a}t}_{EN}}{|\vec{s\hat{a}t}| |\vec{s\hat{a}t}_{EN}|} \right) \quad (5.3)$$

Communication was deemed available when it was at least as large as the configured minimum elevation angle of  $31.73^\circ$ . This resulted in *availability periods* per orbit and surface location, with certain durations between start and end simulation second, and consequent duration changes caused by the Keplerian and circular model w.r.t. SGP4. The communication periods were further characterized by calculating the Euclidian distance between satellite and surface position per simulation second, from which the *propagation delay* was calculated using a radio signal propagation speed of  $299\,792\,458 \text{ m/s}$ , which is the speed of light; as per default configuration for the used INET `radioMedium`.

As is presented in section 5.6 in detail, the comparison of the availability periods of different models showed that generally, their start and end time is shifted in the same direction of simulation time, i.e. there is a shift of the availability period. For each orbit, availability period pairs were formed between SGP4 and the alternative models by periods that overlapped despite this shift. To characterize the shifts independently of the also occurring duration changes of the periods, which influence start and end times, the *zenith time* for each period was identified by its highest

elevation angle, i.e. the shift in time between the periods of a pair are described via the occurring *zenith shift* - it however is only used to illustrate how the availability periods change between models, not as metric to overall characterize them. From the comparison of the individual availability periods, two overlap metrics were calculated: the percentage of period time in SGP4 that is covered by the changed period in the Keplerian and circular model, as well as the ratio of the durations of the period variants in the Keplerian and circular model that lie beyond the SGP4 period to the total duration of the SGP4 period. I.e. in short: (1) *Coverage* - How much of the SGP4 period remains in an alternative model, and (2) how much of the SGP4 duration is not lost, but rather shifted.

While the proposal of this thesis explicitly stated the elevation angle during availability periods, or rather differences of it between the models, as a relevant metric, while working on it it became clear that those differences do not have a meaning for the times when the periods of a pair overlap: Using *space\_Veins* fundamental communication model that only decides on satellite visibility, regardless of the elevation angle beyond the minimum elevation angle, communication is possible. Hence, the elevation angles which matter are just the ones on the edges of the periods of availability period pairs, and the maximum elevation angle from which *zenith/period shifts* are calculated - but differences within the overlap have no meaning and relevance. Thus the elevation angle differences are not generally quantified - their relevant differences are incorporated in the duration changes and shifts of the communication periods.

*Absolute propagation delay differences* were only evaluated in the overlap of the availability period pairs. This is because where they do not overlap, there is a change of availability from SGP4 to the respective alternative model, and delay is only relevant in one of the variants. In addition to the summarizing assessment of all such propagation delay differences as average, the *worst-case delay change* in 6 h of forward and backward propagation, i.e. the highest occurring delay difference of the Keplerian and circular model to SGP4 per subconstellation, was identified.

To evaluate the overall changes of *availability* per (sub)constellation, the changes of the amounts of time where communication with at least one, two, and three satellites was available were calculated.

No metric was evaluated to numerically assess how positional differences and differences in the above communication metrics between the models are linked, but it was visually analyzed if and how *zenith shift* duration changes generally correlate with the identified increase of positional differences with increasing time distance to the TLE epochs.

The **runtimes** were assessed by recording the runtime of the OMNeT++ simulations, as well as of the trace creation for the Keplerian model. More precisely, because

of the parallel execution of the simulations, only the times of actual calculations on the CPUs were used. For the Keplerian model, trace creation and simulation time were added to a total simulation time, because its position precalculations correspond to the position calculations which happen in the circular and SGP4 models during simulation. These absolute runtimes, which are not presented in thesis, (1) were divided by the number of satellites, which allows to assess how runtime scales with the satellite number, i.e. how the time per satellite potentially changes. (2) The absolute runtimes were transformed into ratios of the total runtime in the Keplerian and circular model w.r.t. SGP4. To summarize these metrics over all (sub)constellations, the median was calculated, because significant outliers exist for the (sub)constellations with 9 or less orbits. Generally, this ratio is called *speedup*, which however is a contra-intuitive name considering the massive slowdown resulting from the Keplerian model, which is shown later in the results. (3) Lastly, the ratio of trace creation time to simulation run time was calculated for the Keplerian model.

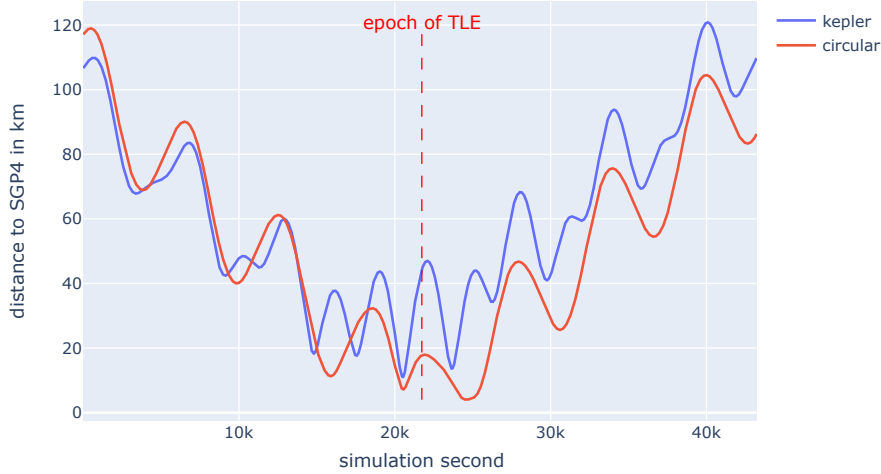
For all the relative change metrics applies that they are more prone to differences between surface locations, alternative models or (sub)constellations, if they are based on small absolute numbers. Similarly, averages and medians applied to the smaller of the considered (sub)constellations, e.g. IridiumNEXTlow or Eccentric, are more prone to deviations. Nevertheless, they were deemed more fit for comparison exactly because of the scaling of absolute numbers with the high variety of orbit numbers in the (sub)constellations. Also important to remember is that the Starlink subconstellation StarlinkLowest, StarlinkLow and StarlinkHigh cannot reach the antarctic surface location due to the inclinations of their orbits.

## 5.6 Results

In the following, firstly the positional differences of the Keplerian and circular model to SGP4 are assessed in detail, after which the communication metrics and the differences therein are evaluated and linked to the positional differences.

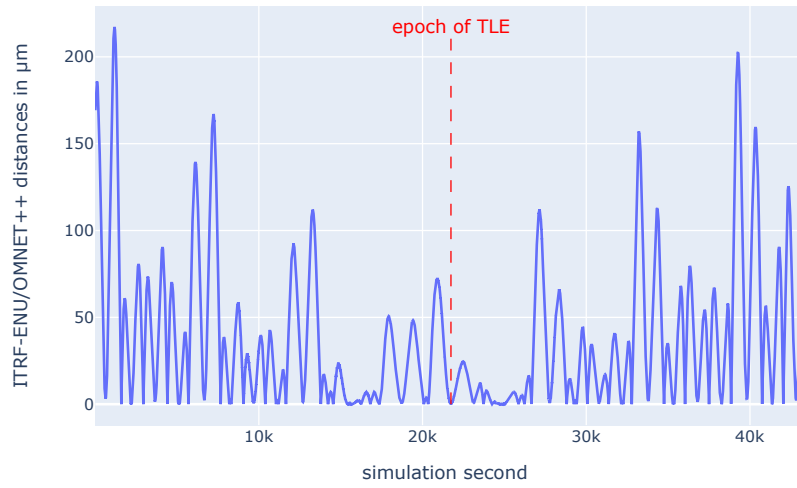
### Positional differences

Before closer assessment of the positional differences between the models, the question if between-model distances are maintained throughout the ITRF-to-ENU/OMNeT++-transformation is examined, using the simulated orbits of the IRIDIUM 128 satellite, which belongs to the *IridiumNEXThigh* subconstellation, and the surface location in Quito as an example. Its between-model distances in OMNeT++-centered ENU coordinates are depicted in figure 5.8, whereas the absolute differences of the between-model distances from ITRF to ENU/OMNeT++ are shown in figures 5.9

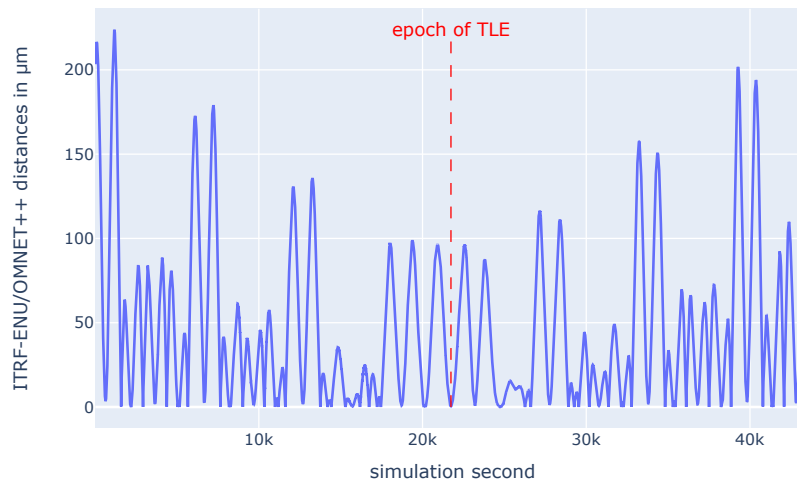


**Figure 5.8** – Euclidian distances of the Keplerian and circular model to SGP4 of IRIDIUM 128 (subconstellation *IridiumNEXThigh*) in ENU/OMNeT++ coordinates w.r.t. the Quito location

(a) and (b). The transformation-induced changes are shown separately because otherwise, they would overlap in a way that limits the visibility for both models. A first insight from these plots is that there actually are changes of the between-model distances by the coordinate transformations from ITRF to ENU/OMNeT++, i.e. they are not exactly maintained – which applies to all orbits of the simulation. The transformation change magnitudes do not directly relate to those of the between-model distances, and most importantly, the magnitudes are very small compared to the between-model distances. The small transformation change - to - between-model distance ratio applies in every case, as is shown by the maximum ratio in the data of all simulated orbits, which is 0.00016 %. It is not clear whether this is based on the theory behind the transformations, caused by floating-point calculations, or caused by both. This finding is sufficient to state that the ITRF-to-ENU/OMNeT++ changes to the between-model distances are negligible and that the distances are roughly maintained. This can be explained by the fact that the between-model distances are already present in the ITRF coordinate system, and exactly the same coordinate transformations were applied to the coordinates between which the distances are calculated in ENU/OMNeT++. Similar effects of the GCRS to ITRF transformation for the Keplerian model, and TEME to ITRF for the circular model and SGP4 can be assumed, as they only introduce Earth rotation to the coordinates; but they were not examined. It follows that the between-model distances can be exclusively examined in ENU/OMNeT++ coordinates hereafter, and that the transformation-induced



(a) Keplerian to SGP4



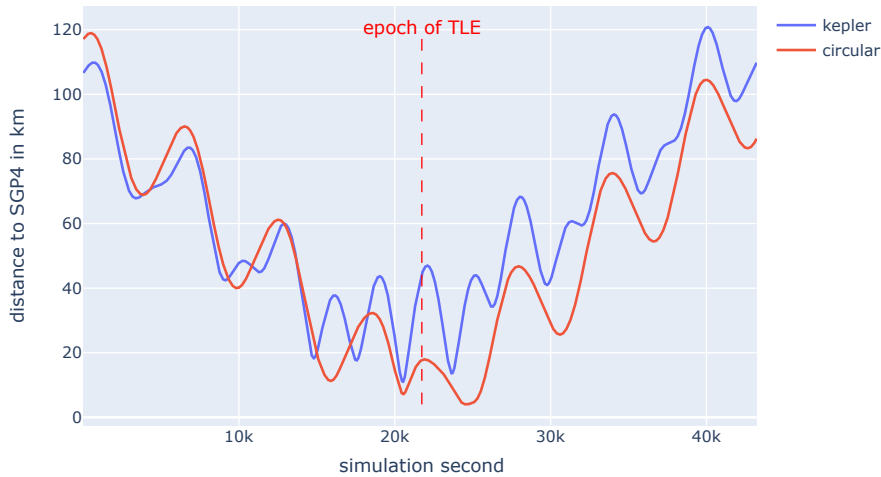
(b) Circular to SGP4

**Figure 5.9** – Absolute differences of the Euclidian distances between the alternative models and SGP4 in ITRF and ENU/OMNeT++

changes are not examined any further.

Now compare figure 5.8 showing IRIDIUM 128's between-model distances in ENU/-OMNeT++ coordinates w.r.t. Quito to the between-model distances w.r.t. Sofia, which are displayed in figures 5.10. They appear to be identical - which also applies to the plot w.r.t. the antarctic position, hence it is not shown. This is a clear indication that the position to which the ENU coordinates are calculated does not change the between-model distances. This can be examined more closely by looking at the whole-simulation-period-sums of positional distances from the Keplerian/circular model to SGP4 of IRIDIUM 128's orbit, which are depicted in table 5.2. In the circular model, the maximum difference between the values is  $36 \mu\text{m}$ , in the Keplerian model it is  $45 \mu\text{m}$  – which are negligible differences considering the magnitude of over 2 million kilometers of the distance sums.

Location-difference-to-between-model-distance ratios of the same scale exist for all orbits in both models, which is why the effect of surface positions and their latitudes on between-model distances is generally deemed negligible, and the between-model distances are only analyzed w.r.t. Quito in the following. Again, it can be assumed that floating-point calculations that use different numbers for different locations are responsible for the differences that do occur, but not proven. The reason for the surface locations having such negligible effects is the same as for the general negligibility of transformation-induced changes stated above, i.e. that the same transformations are applied to each pair of coordinates between which



**Figure 5.10** – Euclidian distances of the Keplerian and circular model to SGP4 of IRIDIUM 128 in ENU/OMNeT++ coordinates w.r.t. the Sofia location



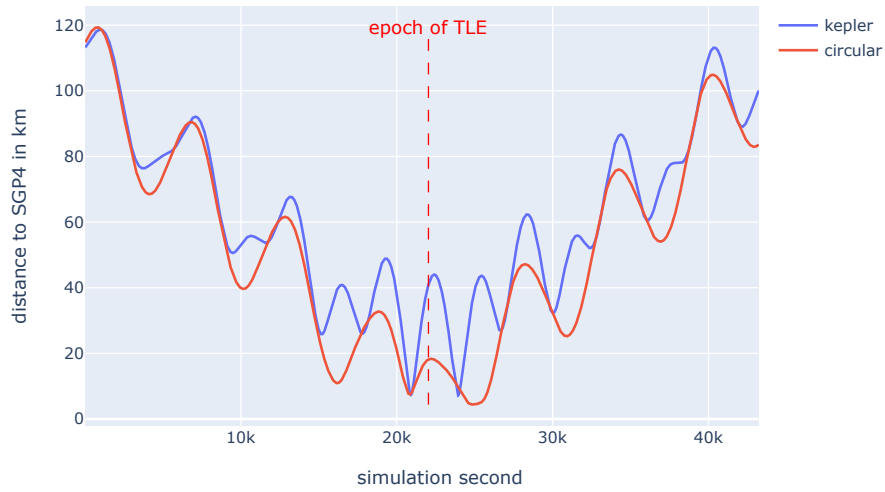
Surface Location	Keplerian to-SGP4 distance sum	Circular to-SGP4 distance sum
Quito	2 676 139.922 766 738 km	2 298 660.140 119 113 km
Sofia	2 676 139.927 667 83 km	2 298 660.140 119 149 km
Antarctica	2 676 139.922 766 777 km	2 298 660.140 119 120 km

**Table 5.2** – Sums of between-model distances over the whole simulation period for the orbit of IRIDIUM 128.

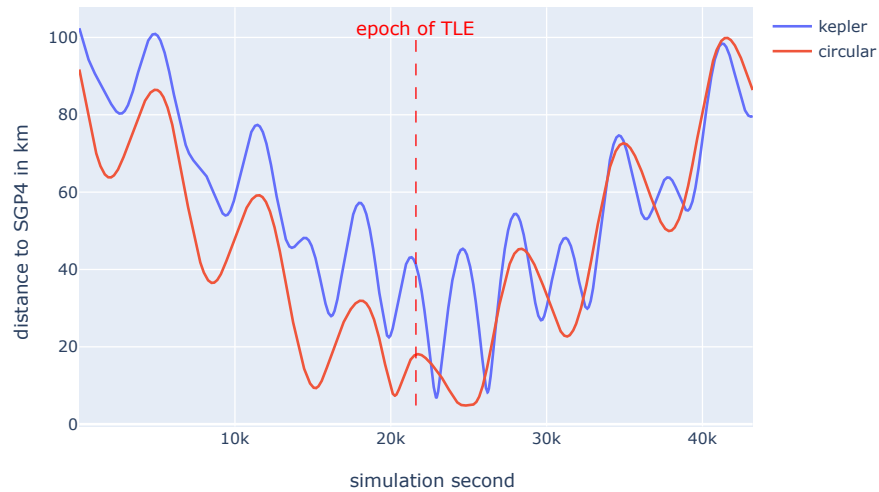
between-model distances are calculated, and this roughly maintains the between-model distances.

For the general evolution of the Euclidian distance of the Keplerian / circular positions to SGP4 positions over time, the simulations resulted in two classes: The positional differences of (1) nearly circular orbits, and (2) more elliptical orbits in the Eccentric constellation. For presenting the general characteristics of between-model distance evolution of the nearly circular orbits, in addition to IRIDIUM 128, IRIDIUM 163 (also in the *IridiumNEXThigh* subconstellation), its between-model distances displayed in figure 5.11, was selected, because their TLE epochs are closest to the median epoch of the constellations which was placed in the middle of the simulation period of each constellation, to show similar periods of backward and forward propagation. For the same reason, ONEWEB 43 and ONEWEB 414 were selected, both in the *OneWebHigh* subconstellation, whose between-model distances are displayed in figures 5.12 and 5.13.

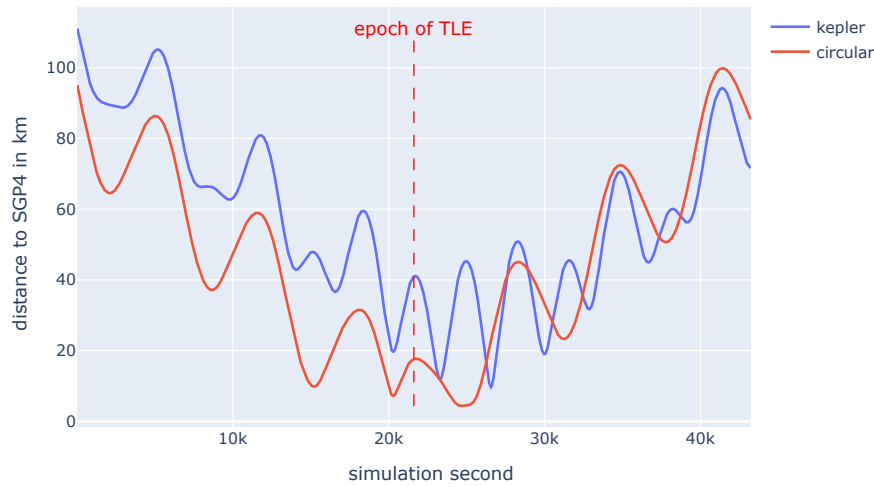
The following properties are showcased which apply to all orbits, also the ones of the *Eccentric* constellation: (1) There is a trend of increasing distances to SGP4 with increasing time distance to the TLE epochs, combined with (2) periodicity of the distances to SGP4. Whereas a period of the circular-to-SGP4 distances consists of one local distance maximum and one local distance minimum, in a period of the Keplerian-to-SGP4 distances there is a second smaller local maximum between the higher local maxima, which however disappears over time. The higher Keplerian local maxima roughly, but not exactly, align with the local maxima of the circular-to-SGP4 distances. In close time distance to the TLE epoch, there generally is a local maximum that aligns with the period of forward propagation. Figure 5.12 makes it clear that this peak can be shortly before or shortly after the epoch. It is roughly at the times of this peaks where the governance of the backward propagation period ends, and the forward propagation period starts.



**Figure 5.11** – Euclidian distances of the Keplerian and circular model to SGP4 of Iridium 163 in ENU/OMNeT++ coordinates w.r.t. the Quito location



**Figure 5.12** – Euclidian distances of the Keplerian and circular model to SGP4 of ONEWEB 53

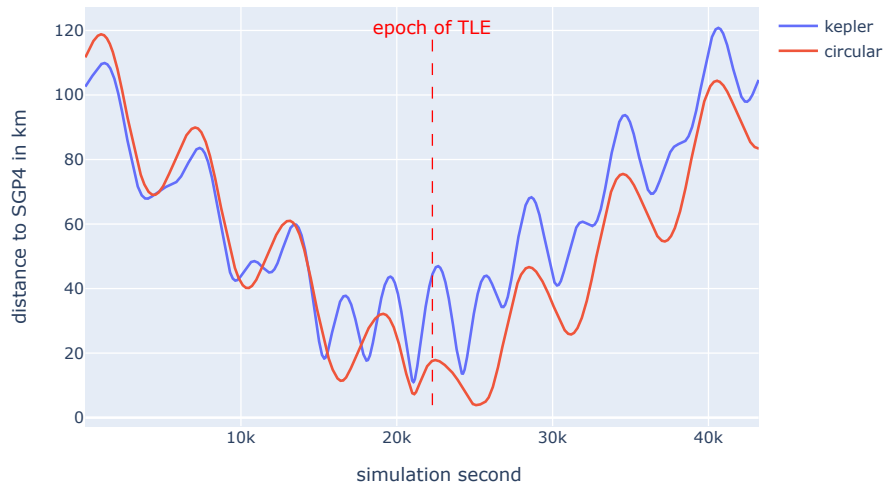


**Figure 5.13** – Euclidian distances of the Keplerian and circular model to SGP4 of ONEWEB 414

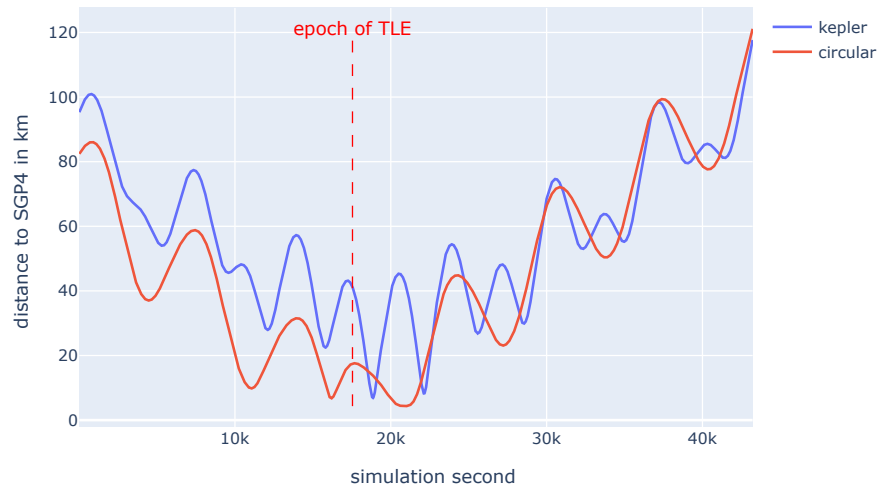
While both (1) and (2) apply to forward propagation and backward propagation, the magnitudes of both regimes differ, as shown especially clearly in figure 5.13. Firstly, consider the Keplerian and circular model comparatively: From the figures, it cannot be stated that one model always yields coordinates more similar to SGP4 than the other in either regime. In the *IridiumNEXT* figures, the similarity to SGP4 of the circular model compared to the Keplerian one worsens from backward to forward propagation, while in the *OneWeb* figures, it gets better. A counterexample for this comparative evolution from backward to forward propagation in *IridiumNEXT* is given by the orbit of the satellite IRIDIUM 107 (too in *IridiumNEXThigh*), whose between-model distances are displayed in figure 5.14, whereas a counterexample in *OneWeb* is given by the orbit of the satellite ONEWEB 17 (too in *OneWebHigh*), whose between-model distances are displayed in figure 5.15.

Fundamentally, the growth of distance to SGP4 in forward and backward propagation can be explained by the common difference of the Keplerian and circular orbit models to SGP4: SGP4 includes various perturbations which are not included in the one-body-problem-based alternative models. Using the equivalent TLE or TLE-derived input parameters, the positions are most similar at the epoch, where the perturbations already cause a positional difference. With propagation from the epoch, further perturbation effects are simulated, or disregarded by the alternative models, and the caused between-model distances add up.

Interestingly, in the vicinity of the epoch in backward propagation and not considering periodicity, the distances to SGP4 do not always increase directly, as shown in figures 5.10 and 5.14. This might be because TLEs are generated using observations



**Figure 5.14** – Euclidian distances of the Keplerian and circular model to SGP4 of IRIDIUM 107 in ENU/OMNeT++ coordinates w.r.t Quito



**Figure 5.15** – Euclidian distances of the Keplerian and circular model to SGP4 of ONEWEB 17

from a period some time before the epoch, which include the perturbations that are also considered in SGP4 simulation, and influence not only the SGP4-exclusive orbital parameters, but also the Keplerian orbital elements used by the other orbit models [5]. So even though they do not consider the perturbations in backward propagation directly, they use parameters which represent the perturbations. However, firstly this cannot be further analyzed because the exact period of the observations is not known [5], and secondly, if this is the reason for not directly growing distances to SGP4, it does not always take effect, as shown by the other displayed between-model distance plots. In those cases, the lack of perturbation simulation in the Keplerian and circular model seems to be much more influential than the TLE adaptation to existing perturbations in the observations before the epoch.

For analysis of the periodicity, the simulation seconds of the local maxima and minima of the circular-to-SGP4 distances were calculated, as the circular model results in only one maximum and minimum per period. For IRIDIUM 163 and ONEWEB 53, these moments in simulation time and the time differences between them of backward propagation are displayed in tables 5.3 to 5.6. Because of the change from backward propagation to forward propagation periodicity, the vicinity

maxima	minima	sim. sec.	time from prev. extremum	time from prev. same type extremum	ITRF z coord SGP4	ITRF z coord circular
1		770	-	-	1416.8	1351.3
	1	4125	3355	-	1009.6	1125.5
2		6793	2668	6023	1365.7	1327.9
	2	10146	3353	6021	1053.8	1141.7
3		12815	2669	6022	1304.6	1315.7
	3	16167	3352	6021	1098.0	1157.8

**Table 5.3** – Backward propagation simulation seconds of circular-to-SGP4 distance maxima and minima of IRIDIUM 163

maxima	minima	sim. sec.	time to next extremum	time to next same type extremum	ITRF z coord SGP4	ITRF z coord circular
	4	24849	3360	6030	1315.7	1319.8
4		28209	2670	6022	1211.9	1257.9
	5	30879	3352	6021	1301.0	1277.2
5		34231	2669	6022	1167.8	1241.8
	6	36900	3353	-	1352.2	1300.6
6		40253	-	-	1123.7	1225.8

**Table 5.4** – Forward propagation simulation seconds of circular-to-SGP4 distance maxima and minima of IRIDIUM 163

maxima	minima	sim. sec.	time from prev. extremum	time from prev. same type extremum	ITRF z coord SGP4	ITRF z coord circular
1		4898	-	-	1456.8	1541.2
	1	8544	3646	-	1008.9	973.1
2		11492	2948	6594	1480.2	1538.1
	2	15138	3646	6594	985.2	976.3

**Table 5.5** – Backward propagation simulation seconds of circular-to-SGP4 distance maxima and minima of ONEWEB 53

maxima	minima	sim. sec.	time to next extremum	time to next same type extremum	ITRF z coord SGP4	ITRF z coord circular
	1	31276	3648	6593	1536.3	1514.4
1		34924	2945	6594	942.8	1014.6
	2	37869	3649	-	1566.7	1518.4
2		41518	-	-	919.1	1017.8

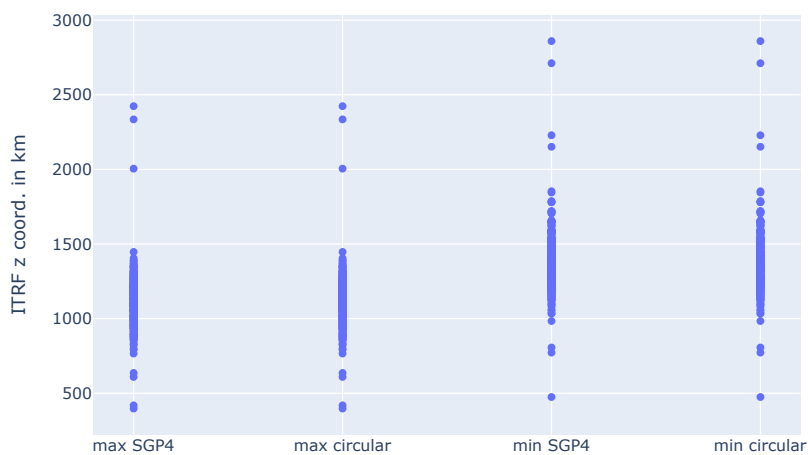
**Table 5.6** – Forward propagation simulation seconds of circular-to-SGP4 distance maxima and minima of ONEWEB 53

of the TLE epoch is not considered. This is why for ONEWEB 53, only 2 periods each were evaluated. As is shown by the ordering of rows according to the simulation seconds of the extrema, backward and forward propagation periods are symmetrical in the succession of maxima and minima. There are variations of just a few seconds in the period time, i.e. the time between extrema of the same type. Overall however, it remains roughly the same - the tables do not show a trend of changing period time. This period is the orbital period, which can be calculated from mean motions  $n$  of the TLEs in *revolutions/day* with  $86400s/n = [s/revolution]$ , resulting in 6024.2 s for IRIDIUM 163 and 6593.6 s for ONEWEB 53. Similarly, considering the backward/forward propagation symmetry, the times between minima and maxima remain the same. For IRIDIUM 163, the time from a maximum to a minimum takes up around 55.7 % of a period, while for ONEWEB 53 it takes up around 55.3 %.

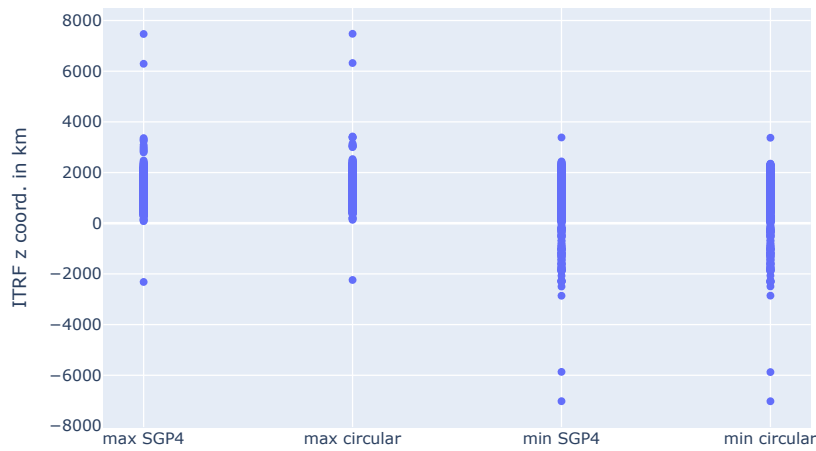
Furthermore, the tables contain the z coordinates of both compared models in the ITRF coordinate system at the time of the extrema, to evaluate if there is a certain value, and a correlated latitude, at which the extrema are situated. Individual analysis of the extrema types shows that there is no fixed z coordinate, but the z coordinate changes. Considering IRIDIUM 163, the z coordinate of the maxima constantly is lowered, and the z coordinate of the minima adapts and is increased. This aligns well with the constant periods between same type extrema and the ones between different types. Also, the z coordinates themselves are intuitive w.r.t. the

ratio of maxima-to-minima and minima-to-maxima periods of the total period. From maxima to minima, both satellites move from a positive latitude around Earth to another positive latitude, and considering table 5.1 both at nearly polar inclinations. On the other hand, from minima to maxima, the satellites seem to stay in the Northern hemisphere, travelling over a smaller distance than the distance from maxima to minima. Considering ONEWEB 53, the coordinate of the maxima first increases with decreasing distance to the epoch, and then decreases again with growing distance to the epoch. These are interesting findings about the nature of the periodicity occurring in the distances of the circular model to SGP4, and they also apply to the Keplerian model. Generally, the periodicity is due to the orbital period. However, while the extrema of the 73 orbits of the IridiumNEXThigh subconstellation actually are grouped around certain z coordinates, as shown in figure 5.16, the extrema of the 599 orbits of the OneWebHigh subconstellation show different groups depicted in figure 5.17. This is likely because IridiumNEXThigh orbits have similar characteristics that determine the extrema z coordinate, whereas in OneWebHigh there are different groups of characteristics.

Unfortunately, the knowledge of this thesis does not allow giving the reasons for the displayed periodicity, and can also not explain the occurring z coordinates and why they change as they do. In the related works, Shuman et al. [48] and Kelso [51] identified oscillations of the distance of SGP4 to the orbits of real satellites which align with the orbital period, and Easthope [49] attributed this to a “primary harmonic term” which is missing in SGP4 w.r.t. a more sophisticated analytical orbit model. However, this thesis only compares the Keplerian and circular model to SGP4 and not to real satellites, so it cannot clearly be assessed if e.g. the Keplerian and



**Figure 5.16** – IridiumNEXThigh ITRF z coordinates of the maxima and minima of the distances from the circular model to SGP4



**Figure 5.17** – OneWebHigh ITRF z coordinates of the maxima and minima of the distances from the circular model to SGP4

circular model are free from oscillations w.r.t. real satellite orbits and the periodicity shown by the to-SGP4 distance plots can fully be attributed to SGP4. Hereafter, no further analysis of the periodicity is performed, which certainly also would require more detailed knowledge of the mathematical details of SGP4.

After periodicity, the growth of distances to SGP4 with increasing time distance to the epoch was examined further. As the growth of the distances seems quite linear, for this, they were fit to a linear function using a least squares method. Because for short fit periods of backward/forward propagation, the concrete instances of periodicity have a high influence on the resulting linear function, and because backward and forward propagation per orbit should be compared, this analysis was limited to orbits for which there is at least 3/8 of simulation time of backward propagation and 3/8 of simulation time of forward propagation, i.e. roughly 4.5 hours each. Also, the average to-SGP4 distance for forward and backward propagation was calculated. This also only used 3/8 of backward and forward propagation each, as if one side has a up to 2/8 of simulation time longer period, its distance averages will certainly be larger. However, this limitation increases the influence of different forward/backward periodicity on the linear fit and the average distances.

For the orbits of the previous plots, table 5.8 shows the results for Keplerian-to-SGP4 distances, table 5.7 for Circular-to-SGP4 distances. Adapting the meanings of the coefficients to this application, from the general linear equation  $y = a + b \cdot x$ ,  $b$  is the growth factor, and  $a$  the distance at the TLE epoch - which because of the fit is different from the actual ones close to a peak of forward propagation periodicity. This is why the following analysis mostly focuses on the growth factor and the average distances.



Satellite	back. growth factor (m/s)	forw. growth factor (m/s)	back. epoch dist. (km)	forw. epoch dist. (km)	back. avg. dist. (km)	forw. avg. dist. (km)
<b>IRIDIUM 107</b>	4.611	4.600	4.292	3.240	41.974	39.205
<b>IRIDIUM 128</b>	4.546	4.477	4.789	4.130	41.999	39.233
<b>IRIDIUM 163</b>	4.579	4.522	4.651	3.932	42.104	39.302
<b>ONEWEB 17</b>	3.851	3.878	5.674	4.017	35.678	35.448
<b>ONEWEB 53</b>	3.463	4.124	8.402	2.361	35.804	35.580
<b>ONEWEB 414</b>	3.538	4.168	8.018	2.207	36.177	35.695

**Table 5.7** – Results of linear fitting the Circular-to-SGP4 distances in backward and forward propagation, where of the general function  $y = a + b \cdot x$  the growth factor is  $b$  and the epoch distance is  $a$ . The average distances result from roughly 4.5 hours of backward and forward propagation.

For the distances based on the circular model, IRIDIUM 107, IRIDIUM 163 and ONEWEB 17 show very similar growth factors in both directions, whereas IRIDIUM 128 has a larger difference between the factors, where backward propagation distances grow faster. However, its differences of the average distances in backward and forward propagation align with the other cases of the IridiumNEXThigh subconstellation where backward and forward growth rates are very similar. The results of ONEWEB 53 and 414 show the forward growth rate being larger with a higher difference than between growth rates of the other orbits. But the average backward and forward distances of the OneWebHigh orbits are all very similar, more than for the IridiumNEXThigh orbits. Considering that the backward epoch distance of ONEWEB 53 and 414 is much larger than the forward epoch distance of the other satellites, the higher differences of those two orbits in both growth factors and epoch distances balance out, which explains why the average forward and backward distances are similarly similar as of the ONEWEB 17 orbit. Consequently, the average distances appear to be a more accurate representation of the difference in distance magnitudes between forward and backward propagation - but they don't provide a quantification over time. In all cases, the average backward distance is larger than the forward distance, but considering the presumed inaccuracies of the fits, a more fitting statement is that for OneWebHigh, forward and backward propagation produce very similar distances, whereas for IridiumNEXThigh, backward propagation produces slightly higher distances than forward propagation.

For the OneWebHigh distances based on the Keplerian model, the backward propagation distance growth is higher than the forward growth in all cases, which correlates to the average distances. For the IridiumNEXThigh distances, IRIDIUM 107 and IRIDIUM 128 have a lower backward than forward propagation growth correlating to the relation between backward and forward average distance. IRIDIUM

Satellite	back. growth factor (m/s)	forw. growth factor (m/s)	back. epoch dist. (km)	forw. epoch dist. (km)	back. avg. dist. (km)	forw. avg. dist. (km)
<b>IRIDIUM 107</b>	3.629	4.117	15.272	25.427	44.274	57.791
<b>IRIDIUM 128</b>	3.560	4.081	15.757	25.695	44.256	57.809
<b>IRIDIUM 163</b>	3.970	3.826	18.203	21.957	50.356	51.848
<b>ONEWEB 17</b>	3.388	3.001	27.844	18.543	54.118	42.312
<b>ONEWEB 53</b>	3.115	3.016	29.602	18.509	54.115	42.316
<b>ONEWEB 414</b>	3.376	2.711	30.529	18.974	57.581	39.685

**Table 5.8** – Results of linear fitting the Keplerian-to-SGP4 distances in backward and forward propagation.

163 is the exception from this correlation, where the backward propagation growth is slightly higher than the forward growth, but the relation of the average distances is vice versa. Analyzing the TLE parameters of the IridiumNEXThigh orbits, these exceptional results were linked to the RAAN, which is the only value which differs considerably in the IridiumNEXThigh subconstellation: The values of IRIDIUM 107 and 128 are close to  $70^\circ$ , whereas the RAAN of IRIDIUM 163 is around  $110^\circ$ . Orbits with very similar RAAN like IRIDIUM 155, 158 and 165 also have average backward propagation distances close to 50 km and forward propagation distances close to 52 km. A RAAN of around  $38^\circ$  seems to be another class with the examples of IRIDIUM 116, 167 and 171, where the average backward propagation distance ranges from 32 km to 40 km and forward propagation from 62 km to 68 km. A closer examination of the RAANs of the ONEWEB satellites showed that ONEWEB 17 and 53 have a RAAN of around  $228^\circ$  and very similar average distances for backward and forward propagation, whereas ONEWEB 414 has one of around  $168^\circ$ . Looking at more examples of these RAAN classes, ONEWEB 407 and 417 with a  $168^\circ$  RAAN also have an average backward propagation distance close to 57 km, and close to 39 km for forward propagation; and ONEWEB 415's and 428's with  $228^\circ$  distances are analogously similar to ONEWEB 17's and 53's. This is an important finding, which however, due to the limited time of a master thesis, could not be more thoroughly examined.

For comparing the circular and Keplerian distances to SGP4, considering the average distances firstly shows that the circular model always produces less average distances to SGP4 than the Keplerian one. Secondly, the larger than / less than relations between backward and forward propagation average distances are not the same using each model for IridiumNEXThigh, but for OneWebHigh. Overall, the differences between backward and forward propagation are reduced in the circular model. Lastly, despite the RAAN classes identified above, the IridiumNEXThigh average distances to SGP4 are very similar in backward and forward propagation

respectively. This also applies when considering the other examples from above. E.g. IRIDIUM 155 is in a RAAN class with IRIDIUM 163, but has a backward propagation average of 41.977 km and a forward propagation average of 39.226 km. Considering the additional examples from OneWebHigh from above yields: ONEWEB 407 and 417 with very similar RAAN to ONEWEB 414 also have average backward propagation distances slightly above 36 km, while forward propagation distances are slightly below 36 km. ONEWEB 415 and ONEWEB 428 with very similar RAANs to ONEWEB 17 and 53 also have average distances more similar to them, with backward propagation average distances of slightly above 35.5 km, and forward propagation distances of slightly below 35.5 km. This very limited analysis of the RAAN classes in comparison of the Keplerian and circular distances to SGP4 indicates that they can be maintained as shown by OneWebHigh, but can also be equalized as shown by IridiumNEXThigh. The comparison of IridiumNEXThigh and OneWebHigh in table 5.1 indicates that this difference is because the OneWebHigh orbits are at higher altitudes than the IridiumNEXThigh ones.

Thus, the findings from tables 5.7 and 5.8 using average distances to SGP4 for backward and forward propagation can be summarized to: (1) The circular model produces less average distances to SGP4. (2) The differences between forward and backward propagation average distances are smaller in the circular model than in the Keplerian one. (3) In the circular model, backward propagation distances are higher than forward propagation distances to SGP4. (4) There appear to be RAAN classes which in the established subconstellations relatively precisely determine the average backward and forward propagation distance. When using the circular model instead of the Keplerian one, they only remain relevant above certain altitudes. One reason for the RAAN being very relevant for determining the distances to SGP4 in backward and forward propagation might be that SGP4 includes perturbations originating from the Sun, and in the short simulation period, the RAAN could determine characteristic positions of the satellites w.r.t. to the Sun. However, this is only a hypothesis that cannot be evaluated without considering how Earth, in addition to its rotation, is oriented w.r.t. to the Sun due to revolving around it.

Now, before explaining them, findings (1), (2) and (3) are to be validated w.r.t. to all nearly-circular constellations, which is done by averaging consideration of the average forward/backward propagation distances to SGP4 of all subconstellations. As the previous orbits showed, the larger/lesser than relation between them can vary within a subconstellation. Thus, the results of distance averaging are divided into two classes: One where the average backward propagation distance is lesser than the forward propagation one, and one where it is larger. For every constellation and subconstellation, table 5.9 for the circular model and table 5.10 for the Keplerian model show for each class: the ratio of the orbit number in a class to the total

Constellation/ Subconstellation	back. to forw. propagation avg. dist. relation class	class instances to total instances ratio	back. avg. dist. (km)	for. avg. dist. (km)
IridiumNEXT	<	0	-	-
	>	1	42.272	39.483
IridiumNEXThigh	<	0	-	-
	>	1	42.055	39.279
IridiumNEXTlow	<	0	-	-
	>	1	44.095	41.189
Starlink	<	0.759	31.704	32.163
	>	0.241	39.274	37.500
StarlinkLowest	<	0.904	38.225	39.060
	>	0.096	42.356	39.975
StarlinkLow	<	0.959	28.645	28.873
	>	0.041	32.469	31.736
StarlinkHigh	<	0.038	32.724	40.749
	>	0.962	35.0286	33.944
StarlinkHighest	<	0	-	-
	>	1	43.754	41.290
OneWeb	<	0.025	32.753	46.745
	>	0.975	36.417	35.921
OneWebLow	<	0	-	-
	>	1	49.861	46.156
OneWebMiddle	<	0.8	32.753	46.745
	>	0.2	44.763	41.907
OneWebHigh	<	0	-	-
	>	1	36.0577	35.650
SatNOGS	<	0.154	46.492	48.089
	>	0.846	48.171	45.810

**Table 5.9** – Circular classes of orbits whose  $\approx 4.5$  hours backward propagation average distance to SGP4 is larger than/smaller than their  $\approx 4.5$  hours forward propagation average distance to SGP4 of all constellations and subconstellations, the ratios of their orbit number to the total constellation/subconstellation orbit number, and the average distances of forward and backward of all orbits of each class. The yellow markings show for which classes in-class average distance comparison is not possible between the Keplerian and circular model because they don't occur in both. The other markings are in support of findings explained in the text.

Constellation/ Subconstellation	back. to forw. propagation avg. dist. relation class	class instances to total instances ratio	back. avg. dist. (km)	for. avg. dist. (km)
IridiumNEXT	<	0.851	43.744	58.733
	>	0.149	56.165	46.615
IridiumNEXThigh	<	0.833	43.571	58.612
	>	0.177	56.165	46.615
IridiumNEXTlow	<	1	44.954	59.586
	>	0	-	-
Starlink	<	0.795	32.882	52.536
	>	0.205	59.004	46.688
StarlinkLowest	<	0.589	47.607	52.281
	>	0.411	61.997	45.370
StarlinkLow	<	1	28.807	51.622
	>	0	-	-
StarlinkHigh	<	1	30.325	57.889
	>	0	-	-
StarlinkHighest	<	0.071	49.053	55.656
	>	0.929	55.551	48.2105
OneWeb	<	0.390	43.553	53.352
	>	0.610	54.944	41.842
OneWebLow	<	1	48.485	56.582
	>	0	-	-
OneWebMiddle	<	0.8	36.6366	63.8755
	>	0.2	61.012	42.013
OneWebHigh	<	0.363	43.739	52.399
	>	0.637	54.883	41.840
SatNOGS	<	0.566	47.758	55.643
	>	0.434	61.490	42.777

**Table 5.10** – Keplerian classes of orbits whose  $\approx 4.5$  hours backward propagation average distance to SGP4 is larger than/smaller than their  $\approx 4.5$  hours forward propagation average distance to SGP4 of all constellations and subconstellations, the ratios of their orbit number to the total constellation/subconstellation orbit number, and the average distances of forward and backward of all orbits of each class. The yellow markings show for which classes in-class average distance comparison is not possible between the Keplerian and circular model because they don't occur in both. The other markings are in support of findings explained in the text.

orbit number, and the average backward and forward propagation distances to SGP4. Because of their accuracy shown above, the growth factor and so-called epoch distance are not part of this evaluation step anymore. The classes where in-class comparison between the Keplerian and circular model is not possible because they do not occur in both are marked in yellow. The other markings are for analysis of the findings (1) to (3). (1) In the backward and forward propagation periods of  $\approx 4.5$  hours, the circular orbit model does not always produce lower distances to SGP4 on average - the circular-Keplerian cell pairs which contain counterexamples are marked in green. However, the 'advantage' of the Keplerian model is only 2.399 km in the upper case, and 3.033 km in the lower case, whereas in the other cell pairs, there are many cases where the circular model causes distances to SGP4 that are more than 10 km lower. Thus overall it can be stated that the circular model is considerably more similar to SGP4 in the examined propagation periods. (2) Only in one case, which is marked in blue, the circular model's backward and forward propagation average distances to SGP4 are more different than those of the Keplerian model. Hence similarly to (1), finding (2) does not always apply, but overall the backward and forward propagation average distances are much more similar than in the Keplerian model. There even are 5 rows for the circular model where the difference is below 1 km, which are marked in orange. This is never the case in the Keplerian model. (3) These results, especially the backward/forward average distance ratios, show that the circular model backward propagation does not always or mostly produces less distances in backward propagation than in forward propagation. The ratios show the same overall property for the Keplerian model. Thus, the results do not allow any general statements on the relation between backward and forward propagation distances. There are subconstellations where larger average backward propagation distances clearly dominate, but also others where this applies vice versa. In the OneWeb and Starlink subconstellations, there respectively are examples for both cases. Only for IridumNEXT the clear relation that smaller backward propagation dominates can be stated - it always occurred in the circular model, and mostly in the Keplerian one. Lastly, no patterns that align with the identified characteristics of the (sub)constellations and their comparison could be identified, e.g. the groups of comparatively increasing inclination from which the Starlink subconstellations were formed, or the rising altitudes of the OneWeb subconstellations.

Findings (1) and (2) in consideration of all nearly-circular constellations can be explained by the fact that by using the average radii from SGP4 simulations of the whole simulation period, the circular model does not only use equivalent inputs to SGP4, but was fit to it. This is not the case for the Keplerian model, which beyond the TLE has no knowledge of the SGP4 satellite positions - this directly explains that finding (1), the circular model's advantage in average distances, almost always applies. That the radius fit is across both backward and forward propagation explains

finding (2), i.e. the approximation of backward and forward propagation average distances in the circular model.

While the above analysis of distances to SGP4 of the near-circular constellations provides a well-funded comparison of the Keplerian and circular model that with average distances abstracts from periodicity, because of the shown inaccuracies of the growth factor from linear fits it fails to characterize the distance growth of the different constellations and subconstellations. To achieve this, worst-case estimates of the distances to SGP4 in forward and backward propagation are provided in table 5.11. For this, the maximum distance to SGP4 that occurs in backward/forward propagation periods of half of the simulation period, i.e. around 6 hours, was identified. Under the assumption that the growth of the distances to SGP4 remains

Constellation/ Subconstellation	Alt. model	back. dist (km) 6 h	back. dist (km) 24 h	forw. dist (km) 6 h	forw. dist (km) 24 h
IridiumNEXThigh	Circular	120	478	105	422
	Keplerian	127	507	129	516
IridiumNEXTlow	Circular	122	487	107	427
	Keplerian	114	457	129	517
OneWebHigh	Circular	105	419	126	505
	Keplerian	111	444	118	470
OneWebMiddle	Circular	140	560	-	-
	Keplerian	134	537	-	-
OneWebLow	Circular	141	565	137	550
	Keplerian	133	532	120	480
StarlinkHighest	Circular	123	493	108	430
	Keplerian	149	594	118	472
StarlinkHigh	Circular	106	423	111	442
	Keplerian	89	356	142	570
StarlinkLow	Circular	119	478	116	463
	Keplerian	92	367	144	574
StarlinkLowest	Circular	152	607	132	527
	Keplerian	151	603	152	609
SatNOGS	Circular	604	2414	403	1614
	Keplerian	252	1009	203	813

**Table 5.11** – Worst-case distances of the circular and Keplerian model to SGP4 in 6 hours of backward and forward propagation, which are extrapolated to 24 hours of propagation each. Worst-case means that the highest distance reached between an orbit of SGP4 and the respective other model per (sub)constellation is displayed. The cells marked in yellow are the highest values of the constellation in the circular and Keplerian model respectively. The OneWebMiddle subconstellation did not contain any orbits which were propagated for 6 hours after their epoch.

as linear as it was shown in the previous plots, and also by all the other ones that are not shown here, this is extrapolated to a day. i.e. multiplied by 4. Where subconstellations exist, only their estimates are shown, as the highest forward or backward estimate determines the estimate for the constellation. These highest backward/forward estimates per constellation are marked in yellow. Note that rounded values are displayed for both 6 hours and 24 hours, but the original 6-hour values were multiplied by 4. With the extrapolation to 24 hours, the worst-case distances can be related to the distance of SGP4 to real satellite orbits after one-day propagation, which is 20 km based on related work in chapter 3. Under another worst-case assumption, w.r.t. to real satellite orbits, the worst-case distances simply add up with the SGP4 maximum distance, i.e. the worst-case distance to real satellite orbits is given by the depicted values plus 20 km. As the related work on SGP4 did not cement the linear growth of its distances to real satellite orbits, the estimate should not simply be divided by 4 to combine it with the worst-case to-SGP4 distances of the 6 h propagation periods.

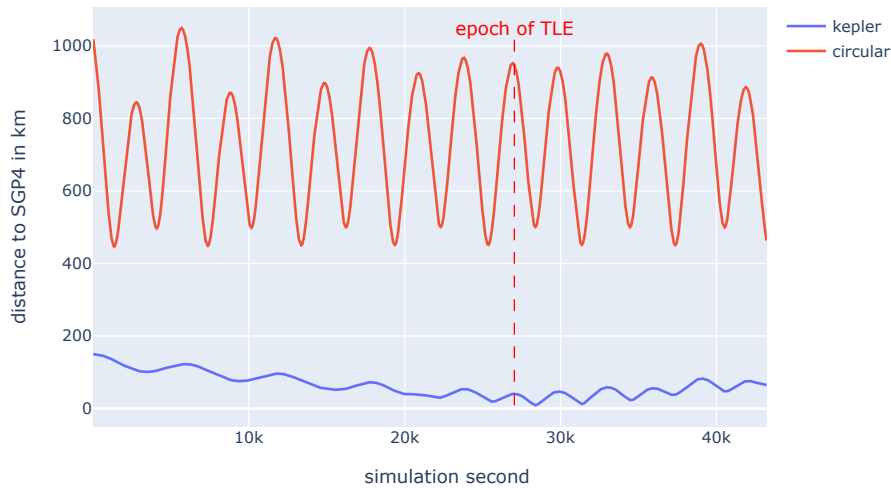
Presumably because the data of this table characterizes the constellations by single to-SGP4 distance values of individual orbits - in contrast to the previously used averages - the findings that the circular model generally produces less distances to SGP4 than the Keplerian one, and that its forward and backward distances are more similar than of the Keplerian one, are not reflected here. For the same reason, the displayed distances are generally not linkable to the identified characteristics of the constellations and subconstellations. However, SatNOGS has exceptionally high values in comparison to the other models, especially when simulated with the circular model. This is because the displayed circular worst-case distances belong to the JAS-2 (FO-29) satellite orbit, which has an eccentricity of 0.035 - which is very close to the limit for being considered a nearly circular orbit, and more similar to the eccentricities of the Eccentric constellation than to those of Starlink, IridiumNEXT and OneWeb. Analogously for forward propagation, the orbit belonging to the shown worst-case values belongs to MAKERSAT 0 with an eccentricity of 0.021, even though this is not quite as close the Eccentric constellation eccentricities. It was already hypothesized before the simulation study that the circular model would perform worse for such orbits than for rather elliptical ones, and the analysis of this is completed by examining the Eccentric constellation in the next paragraph. The noticeably higher worst-case distance of forward propagation in application of the Keplerian model to SatNOGS compared to the other constellations belongs to the orbit of satellite LAPAN-A2, which has no remarkable TLE parameters or remarkable average radius. Similarly, the Keplerian backward worst case for SatNOGS is worse than for the other constellation, and belongs to the orbit of satellite GLOBAL 7, which has a relatively low average radius and high mean motion. This is one indicator that lower radii and higher mean motion, i.e. faster angular speed on the orbits,



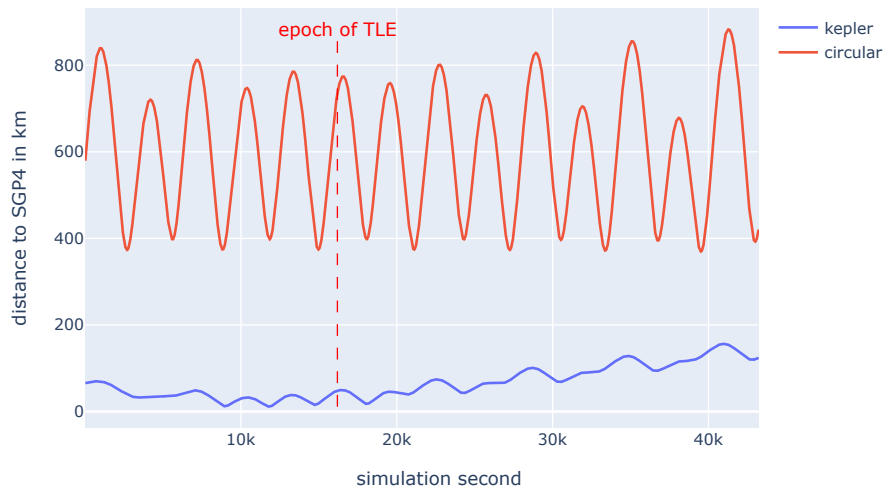
may lead to higher distances to SGP4. However, neither table 5.11 nor the previous tables support generalizing this, e.g. because in the currently considered table, the worst cases of StarlinkHigh compared to StarlinkHighest are significantly smaller in backward propagation. Also for the worst-case estimates there is no general relation between backward and forward propagation distances.

Hence, excluding the circular SatNOGS case, the table can be summarized to: Assuming linear growth of distances with increasing time distance to the epoch, which is a reasonable assumption, and consequent extrapolation, in the worst cases of the nearly circular orbits, the distances excluding SatNOGS values reach up to 427-609 km after one day of propagation, while SatNOGS' worst cases of the Keplerian model are exceptional with 813-1009 km. Using the maximum estimate of SGP4 distance to real satellite orbits, and the worst-case assumption of them adding up, this maps to 447-629 km and 833-1029 km distances to real satellite orbits respectively after a propagation period of a day. SatNOGS' worst cases in the circular model can be attributed to modeling relatively elliptical orbits as circles.

This effect shows even more when considering orbits of the Eccentric constellation. The between-model distances of the two representative orbits with an epoch closest to the constellation median epoch, POPACS 3 and PROSPERO (BLACK ARROW) are depicted in figures 5.18 and 5.19. In contrast to the between-model distances of all other constellations, here the circular model generally performs much worse than the Keplerian one. The fundamental properties identified previously are still visible: The distances to SGP4 grow with increasing time distance to the epoch in forward and backward propagation, i.e. the increasing influence



**Figure 5.18** – Euclidian distances of the Keplerian and circular model to SGP4 of POPACS 3 in ENU/OMNeT++ coordinates w.r.t Quito



**Figure 5.19** – Euclidian distances of the Keplerian and circular model to SGP4 of PROSPERO(BLACKARROW)

of the perturbations included in SGP4 but not the other models still applies here. Also periodicity is shown again, aligning with the orbital period. Interestingly, the circular model now shows two local maxima in the orbital period. However, as stated before, this thesis refrains from further analysis of periodicity.

		Circular	Keplerian
back. > forw.	ratio	0.667	0.333
	avg. back. dist. (km)	679.766	30.211
	avg. forw. dist (km)	694.660	50.117
back. < forw.	ratio	0.333	0.667
	avg. back. dist. (km)	739.992	57.581
	avg. forw. dist (km)	718.830	34.672
worst case back. (km)	6h	1690	122
worst case forw. (km)	6h	1078	128

**Table 5.12** – Eccentric to-SGP4 distance metrics comparable to tables 5.9, 5.10 and 5.11

The metrics characterizing backward and forward propagation distances to SGP4 for the whole Eccentric constellation are shown in table 5.12. Due to the small number of orbits in the constellation, the required simulation period available both for backward and forward propagation of an orbit for calculating the average distances and assessing their relation had to be adapted to 2/8 of the simulation period, i.e. around 1.5 hours. However, this still only allowed to consider 3 orbits. The period for worst-case distances in backward and forward propagation remained at roughly 6 h. Here, 2 of the 8 orbits were excluded because their simulation period does not reach the epoch. Extrapolation was not applied because the periodicity showcased by the circular model clearly dominates the to-SGP4 distance magnitude in comparison to the presumably linear growth with increasing time from the epoch. The circular average distances to SGP4 are clearly much higher than all the ones in tables 5.9 and 5.10, and also its worst case distances than the worst case distances in table 5.11 through the circular modeling of elliptical orbits. The Keplerian average and worst-case distances are generally comparable to those in these tables, showing that the Keplerian is capable of modeling positions on more elliptical orbits with similar fidelity. As in previous cases, there was no clear advantage between backward and forward propagation for the Keplerian or circular model, the shown ratios should not be generalized for more elliptical orbits as in the Eccentric constellation.

As the related works provide assessments of the positional accuracy of the Keplerian and circular model, the overall findings of this section can lastly be compared to them. Considering Kast's [41] work, which is the only one implementing and evaluating a circular model, it must be stated that his results in comparison to very precise satellite positions are clearly superior, with e.g. an along-track error of only 65 km even after 7 days of forward propagation. This is likely because his circular orbits were modeled by generally fitting them to the ground truth orbits, whereas the implementation of this thesis mostly is based on TLE parameters and only fits the radius to the SGP4 reference orbits. Because the intention of the models implemented in this thesis is to provide solutions for simulating artificial orbits, which is possible with the circular model that not necessarily requires a TLE - a radius can also arbitrarily be specified - such a generally fit-based approach is not a solution for increasing the accuracy of the circular model. For the Keplerian model, only a low estimate of positional error growth in forward propagation is provided by the Monte-Carlo simulation of Morales et al. [39], which propagates the satellite from a new truth position in each time step - hence this is not a very meaningful comparison. Nevertheless, applying the growth estimated from their findings,  $\frac{1\text{m}}{450\text{s}}$ , to the 6 h period for which the above forward worst-case estimates were calculated, yields an estimated error of around 48 km w.r.t. GPS-derived truth satellite positions. This is less than half of all Keplerian forward worst-case estimates w.r.t. to SGP4, i.e. even without adding

---

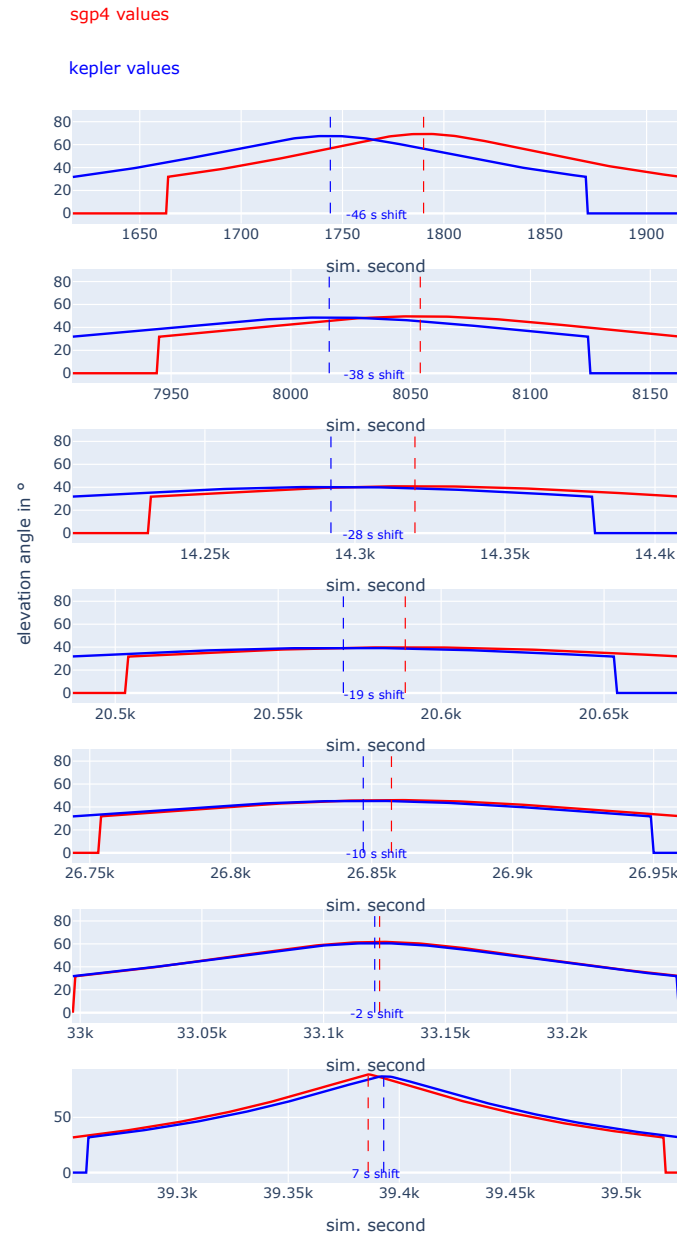
any SGP4-to-real-satellite distance. Considering the difference between worst-case and low Monte-Carlo estimate, this at least does not seem unreasonably different.

### Availability periods

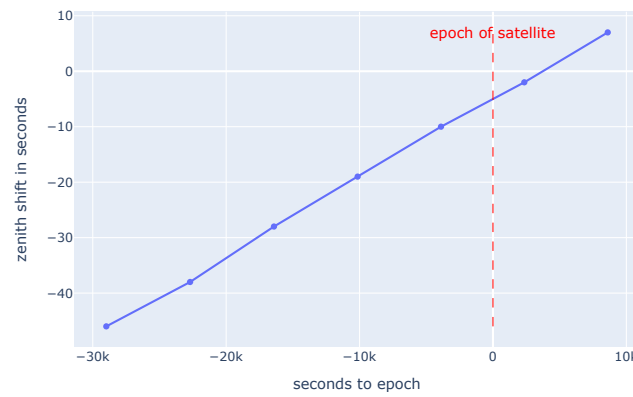
In the following, it is presented how the use of the Keplerian and circular orbit model changes the periods of communication availability determined by the minimum elevation angle w.r.t. SGP4. Firstly, an example orbit w.r.t. an example location showcasing effects generally observed in the results is considered. Consider figure 5.20 showing the elevation angles and resulting availability periods of the SatNOGS satellite LAPAN-A2 w.r.t. the Quito location, which, from top to bottom, get continuously closer to its TLE's epoch at simulation second 30650:

From the first plot to the last plot, i.e. with decreasing distance to the epoch, the degree to which the availability periods of the models are unaligned decreases. This is clearly indicated by the dotted vertical lines marking the elevation angle zeniths of the respective periods - the time distance between them decreases with each communication period, and in the last plot very close to the epoch, the zeniths of the periods are almost fully aligned. This occurs generally for all orbits in comparison of the models, in backward as in forward propagation, and correlates with the increasing positional differences between the models with increasing time distance to the TLE epoch. Another general property showcased here is that the period shift dominates the differences between the periods. Their durations change too, but this is only visible in the figures with a closer look at the respective starts and ends of the periods, before/after which the plotted (but not calculated) elevation angle is  $0^\circ$ , to clearly mark these points in time: Especially considering the plot at the very bottom, the difference between the period start times and period end times is different. The period shift has a very similar length to these differences, showcasing that they are mostly caused by a shift of period start and end time together with the time of the maximum elevation angle w.r.t. the SGP4 period.

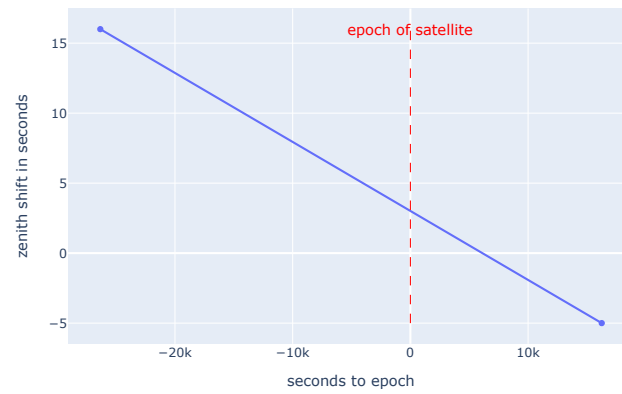
As the positional differences between the models are absolute values, they increase in both the direction of backward and forward propagation. The zenith/period shift, which can have negative or positive values, behaves differently. The values of the zenith shifts of LAPAN-A2 in relation to the simulation time are depicted in figure 5.21 and (1) show that it increases in forward propagation, and (2) has the potential to be very closely approximated by a linear function of time distance to the TLE epoch in one direction of time, i.e. analogously as a function of simulation time. Also this linearity is a general property of the results. However, the linear shift behavior shown is not always one of monotonic growth, but can also occur in reverse form as monotonic decrease in the direction of simulation time. This is shown by the analogous plot of the ONEWEB 456 satellite in figure 5.22. Representing only two availability periods, it fundamentally is not the best example one could hope for, but LAPAN-A2 is the only orbit which had that many communication periods w.r.t. Quito, and in fact, during the rather shorter simulation period even orbits with two



**Figure 5.20** – Elevation angles and availability periods resulting from it of the SatNOGS LAPAN-A2 satellite's orbit w.r.t. Quito comparing the Keplerian model to SGP4. Where the elevation angle is below the minimum elevation angle of  $31.73^\circ$ ,  $0^\circ$  is plotted to clearly distinguish the positions of the availability periods in simulation time.



**Figure 5.21** – Zenith shifts of LAPAN-A2 in the Keplerian model w.r.t. Quito and SGP4



**Figure 5.22** – Zenith shifts of the ONEWEB 456 w.r.t. Quito

availability periods w.r.t. Quito were rare. Nevertheless, ONEWEB 456 shows how the apparent linear zenith function can also be negatively inclined. It was found that while shift magnitudes differ between the Keplerian and circular model, the time direction in which the shifts increase is equal. Zenith shifts of a subconstellation do not generally grow in the same time direction. Reasons for this that might be found in the respective orbital parameters could not be analyzed due to time constraints.

The insight that reduced overlaps between the availability periods in the compared models are often mostly due to the zenith shift, while the change in period duration often is relatively small and less responsible, was the reason for not only calculating the overlapped percentage of the SGP4 period, i.e. *Coverage*, but also the ratio of the amount of time in the Keplerian and circular model that does not overlap with SGP4 to the period duration in SGP4, as it represents how much of the SGP4 period is not lost in an alternative model, but shifted. E.g. if only 60 % of the

SGP4 period overlap with the period in an alternative model, but the duration of the non-overlapping part is 40 % of the SGP4 period, the period duration is fully preserved. If one of the periods is larger or smaller than the other, the overlap time plus this ratio will be larger or smaller than 100 %. In other words is this sum the relative duration change from SGP4 to the Keplerian or circular model. SGP4 periods which are not overlapped by a communication period of the same orbit in an alternative model simply have a *Coverage* of 0. These metrics were averaged for each subconstellation.

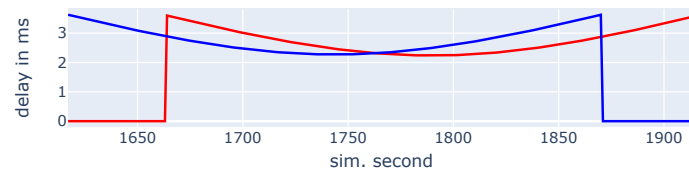
The comparison of the Keplerian model and SGP4 resulted in median 93.07 % of overlap and median 3.24 % change of the availability period durations w.r.t. Quito, 93.36 % of overlap and 6.45 % change of the durations w.r.t. Sofia, and in 94.38 % of overlap and 6.45 % change of the durations w.r.t. the antarctic location. The results of the circular model are: 94.59 % overlap and 3.15 % duration change w.r.t. Quito, 92.35 % overlap and 5.94 % duration w.r.t. Sofia and 95.07 % overlap with 2.51 % duration change w.r.t. the antarctic position. Thus, generally no model significantly outperforms the other. However, the circular models weakness in approximating more elliptical orbits shows for the Eccentric constellation: 55.31 % overlap and 56.56 % of duration change w.r.t. Sofia, 64.06 % o./ 39.97 % d. change. w.r.t. Antarctica and 71.72% o./ 24.81. d.c. w.r.t. Quito. As these indicator metrics of period shifts and duration changes do not scale with the latitude of the respective locations, the variety of values is likely due to the different orientations of the only 8 relatively elliptical orbits in the constellation, and how the circular model fails to approximate availability periods especially badly w.r.t. Sofia. Their individual orientations however were not assessed. Another notable result is StarlinkHighest with only 27.44 % overlap but 151.60 % of duration change w.r.t. Sofia in the circular model. The existence of overall 20 availability periods of SGP4 in the subconstellation which are not covered by an overlapping period at all, 9 more unmatched ones in the circular model, as well as a 296.84 % duration increase w.r.t. to Kunlun without any overlapping periods indicates that (a) the subconstellation contains most of the TLEs of the Starlink constellation whose epoch is quite far from the simulation start, or (b) this is due to their high velocity at lower altitudes, or (c) because of orbital parameters not considered when characterizing the subconstellation, like RAAN or AOP, or (d) a combination of multiple such effects. (c) is the most likely cause, because in epoch times and velocities, StarlinkHighest is not as extremely different from the other subconstellations as the results would imply. Another, complete failure of the circular orbit model to recreate the availability periods of SGP4 is presented by SATNOGS w.r.t. Sofia and Antarctica, whereas the model performs well w.r.t. Quito with 90.11 % overlap: There are no overlaps with the availability periods for Sofia, with a duration change of 48.72%, and also no overlaps with



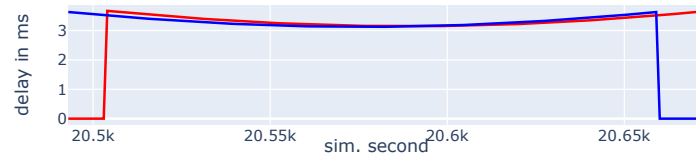
Kunlun, where the duration change is only 14.39 %. This in comparison small duration change w.r.t. Antarctica firstly would imply that the availability periods to there existing in SGP4 are mostly shifted too far in time - 3010 communication periods from SGP4 were not matched. However, the circular periods of SatNOGS that do not overlap with any of SGP4 are only 310 in number. Consequently, it can be stated that the circular model causes many of the orbits to deviate that far from the polar positions that many shorter periods are lost - but other ones are created, preserving a lot of communication duration. This is certainly helped by the many high deviations of SatNOGS orbit epochs from the median epoch calculated for it and the consequent simulation start time; and the higher eccentricities below the 'nearly circular' limit occurring in the constellation.

Consequently, it can be concluded while there are many cases where the circular orbit model manages to maintain a lot of the availability periods and their duration, especially w.r.t. to Sofia and Kunlun there are, aside from the Eccentric constellation which is problematic in the circular anyway, certain conditions that lead to utter inaccuracy in these metrics. In contrast to this, produces the Keplerian model overlaps above 90% and duration changes below 5 % for almost all (sub)constellation-surface location combinations. Thus, on the current level of knowledge on the characteristics of scenarios that lead to high inaccuracies of the circular model, it is too arbitrary in its duration changes to be a generally valid alternative to SGP4 for scenarios which depend on certain communication durations. As certain availability requirements like 1-satellite availability might still be provided by it, availability is discussed overall at the end of the next section. On the other hand, the Keplerian model causes relatively expectable changes which lead to durations that are still similar to the ones of SGP4. So if not certain durations are required, it seems to be a valid alternative model for now.

Using LAPAN-A2 again, one example of the effects of period shift and duration change with general applicability is given for the Keplerian model w.r.t. to SGP4 and Quito, depicted in figure 5.23, which shows the same overlapping periods as at the top of 5.20. Similar as the evolution of elevation angles of LAPAN-2, the



**Figure 5.23** – Delay between shifted communication periods of the LAPAN-A2 satellite's orbit. Outside of the availability periods cannot be communicated, hence the delay is 0.



**Figure 5.24** – Delay between less shifted communication periods of the LAPAN-A2 satellite's orbit. Outside of the availability periods cannot be communicated, hence the delay is 0.

evolution of delay in the periods is very similar. The differences of delay where the periods do not overlap were not calculated, as this would only represent the fact that communication is possible at one point of time when it is not possible in the other - which already is represented by previously described metrics. By estimating from the figure, even the maximum delay difference in this case is only 1 ms. A period overlap of LAPAN-2 closer to its epoch, depicted in figure 5.24, which corresponds to the elevation angle plot directly in the middle of figure 5.20, shows how much more similar delays are when the shift between the availability periods is smaller.

Thus, the assessment of the average delay changes between overlapping availability periods of a subconstellation produced mostly very small values. Commonly, they are below  $30\ \mu\text{s}$ . Again, the Eccentric constellations could be identified as outliers, for the circular model and w.r.t. with Quito with a 0.5 ms average, and here even for the Keplerian model with a 0.1 ms average. This is likely due to the simulation start being 6 hours before the median epoch, where the epochs of a larger subset of the low number of 8 orbits is relatively time-distant to the current simulation time for a considerable duration of the simulation, and thus the availability periods are considerably shifted to each other. W.r.t. applications which do not depend on constantly very low delay, such average changes do not reduce the validity of the circular and Keplerian model as alternative for SGP4. For assessing the validity w.r.t. real-time applications requiring constant very low propagation delays, the maximum instantaneously occurring delay increase in 6-hour backward and forward propagation from the TLE was identified. These values are generally very similar for the Keplerian model and the circular model, where a lower limit is given by the StarlinkHigh subconstellation with around 3.35 ms, whereas the upper limit is given by SatNOGS at around 8 ms. As this is only an increase of one-way propagation delay adding up to existing delay, real-time capabilities could be considerably hindered by such a change. However, the validity of the alternative models to space\_Veins w.r.t. propagation delay depends on the specific real-time application and how it can tolerate instantaneous deviations.

### Overall availability

To assess the availability across all orbits of each subconstellation, and constellation where there are no subconstellations, the ratio of how much the amounts time changed in which (1) at least one, (2) at least 2, and (3) at least three satellites were available, was calculated. For assessing the accuracy of availability modeling, positive and negative deviations from SGP4 were considered equally relevant and thus considered as absolute percentual changes instead of increases / decreases.

Similar to previous metrics, both the circular and Keplerian model produce very similar availability time amounts to SGP4 for many location-constellation combinations. Thus, the circular median ratios of 1-satellite availability change are Quito 0.95 %, Sofia 1.2 %, Antarctica 2 %, whereas the Keplerian model shows Quito 1.8 %, Sofia 1 %, Antarctica 1.4 %. Even lower median changes show for all cases of 2-satellite availability, except for the circular model at Quito, where the median change rises to 2.9 %. 3-satellite availability proves to be much more volatile, with median changes larger than 10 % in all cases. The most significant outliers are: The Eccentric constellation, which due to its orbit number only allows consideration of 1-satellite mobility. The circular model changes for it reach from 7.8 % to 29.6 %, whereas it does not have significant effects combined with the Keplerian model. Also again, aside from Eccentric, the SatNOGS and StarlinkHighest constellations are affected the most by the circular model when being applied to the Sofia and Antarctica locations: StarlinkHighest's 1-satellite availability changes by 46.4 %, and 2-satellite availability even by 122 % w.r.t. Sofia, while SatNOGS has the 3rd-highest 1-satellite availability time amount change with 10.9 %. These changes can be explained by the previously given characteristics of the constellations. In contrast, for all constellation-location combinations are the 1-satellite and 2-satellite availability changes due to the Keplerian model smaller than 6 %. However requiring 3-satellite availability also enables the Keplerian model to introduce major changes to the availability time, e.g. with 140 % of change. Assessment of the absolute time amounts of three-satellite availability in SGP4 showed that its proneness to change is mostly due to the small reference time amounts. There also are positive outliers: The StarlinkLow subconstellation, which cannot communicate with the antarctic location, seems to be relatively immune to changes with values consistently below 4 % w.r.t. Quito and both the Keplerian and circular model, and even changes lower than 1 % for Sofia. Even more immune to change is the OneWebHigh subconstellation, where no change exceeds 0.5 %, and many changes were even rounded to 0 % with the used precision. This can be mostly explained by them having very low eccentricities, and the fact that they are large planned constellations with requirements to coverage and availability. Conclusively it can be stated that the Keplerian model consistently achieves relatively low changes to 1- and 2-satellite availability, whereas for the

circular model the already previously established problems with elliptical orbits and certain other constellation-location combinations also show here. This aligns with the findings on the overlap of availability periods in the previous section. Thus, when only 1- or 2-satellite availability similar to SGP4 is required, the Keplerian model can be generally used as alternative model with relatively small and expectable changes to availability. However, the Keplerian changes to 3-satellite availability can be very considerable and it is not a valid replacement to SGP4 in this regard. The invalidity of the circular model as a general alternative to SGP4 is manifested by these results, and because the reasons for its deviations could not be generally identified, it should be expected to also considerably deviate from SGP4 for other conditions than which were already identified here.

### Runtime

For the assessment of runtimes, firstly, the ratios of the total runtimes of the circular and Keplerian model w.r.t. the SGP4 runtimes are compared, i.e. for the Keplerian model including position precomputation in trace creation in Python. The median ratio of total runtime to SGP4 of the Keplerian model is 13.667, whereas the median ratio of the circular model to SGP4 is 0.934. From this follows that the ITRF trace precomputation in Python and the subsequent reading of the traces during C++ simulation in the Keplerian model is much more time-consuming than the calculation of ITRF satellite positions from SGP4 that is purely in C++. Because of the use of different languages for satellite position calculations, this is just a statement on the relative speed of the implementations, but not generally on the speed relation between Keplerian and SGP4 calculations, which requires evaluation in the same language. The employed Python library *poliastro* only uses Python and no underlying other language [54], which tends to perform worse than C++ in astrophysical calculations as shown by Akeret et al. [65], so the high ratio w.r.t. SGP4 can actually be attributed to Python. However, as also shown by Akeret et. al, speeds for astrophysical computations comparable to C++ can be achieved with a dedicated Python compiler. *Poliastro* employs such a compiler, but not the one used by Akeret et. al, so one can assume that the speed could potentially be improved even using Python. From the between-language comparisons in their work also follows that fully implementing Keplerian calculation, or at least the transformation from GCRS to ITRF, in C++ would probably also improve the Keplerian runtime. This would affect the total runtime significantly, because currently, the median ratio of trace creation time to simulation time is 16.064. Because the current implementation of trace creation always computes traces for a set of TLEs given in one file sequentially, the burden introduced by it can be reduced for cases where multiple CPUs are available w.r.t. one TLE file by implementing parallel processing capability.

---

As the circular model runs fully in C++, it can be assumed that it generally leads to faster calculations than SGP4, which makes sense as its calculations are less sophisticated than the ones of SGP4. Of course, the numerical findings in this regard depend on the actual implementation of this work, so they do not allow an absolute statement on the models' relation. In both the Keplerian and circular model, the runtime per satellite was very similar for all (sub)constellations. Because the satellite numbers ranged from 7 to 908 satellites per simulation, it thus can be stated with high certainty that the models' runtimes scale linearly with the satellite number.

---

## Chapter 6

# Conclusion

---

To explore alternative models of satellite mobility enabling the creation of artificial LEO satellite orbits for the simulation framework `space_Veins`, which currently employs SGP4 that models only existing orbits, this thesis implemented (1) satellite movement on orbits based on Johannes Kepler's two-body problem, and (2) a simplified variant of it which models all orbits as circular. To assess if they present valid alternatives to `space_Veins`, their deviations from SGP4 in satellite positions and consequent communication metrics of propagation delay and availability were evaluated and linked, by representing constellations of existing satellites in the models via equivalent inputs. The comparison of the positions calculated from the models showed that when modeling nearly circular orbits, the potential alternatives deviate similarly from SGP4, but the circular model yields much higher position errors when orbits are more elliptical. In evaluation of periods of availability w.r.t. three Earth surface locations at different latitudes, availability changes were linked to increasing positional deviations between the models as well as to certain constellation characteristics such as the number of simultaneously simulated satellites, and quantified per constellation-location combination. The deviating availability periods were further characterized by the propagation delay changes in comparison to SGP4. This concluded in the judgement of the circular model's potential deviations to SGP4 as too substantial, and the rejection of its validity as a generally usable alternative orbit model. In contrast, the Keplerian model was shown to calculate reasonably similar communication characteristics with comparatively expectable deviations, and thus is deemed a valid alternative. However, the relevance of its deviations from SGP4 for various applications has yet to be evaluated, for example the effects of instantaneously occurring maximum propagation delay increases on real-time communication applications. Furthermore, this thesis could only superficially analyze the variety of orbital characteristics in the employed satellite constellations and the influence of the surface locations. Additional research on this would provide

a clearer understanding of under which circumstances certain deviations of the Keplerian model w.r.t. SGP4 are to be expected, or how they potentially could be avoided.

---

## List of Abbreviations

---

<b>AOP</b>	Argument of Periapsis
<b>ECEF</b>	Earth-Centered Earth-Fixed
<b>ECI</b>	Earth-Centered Inertial
<b>ENU</b>	East-North-Up
<b>GCRS</b>	Geocentric Celestial Reference System
<b>ITRF</b>	International Terrestrial Reference Frame
<b>LEO</b>	Low-Earth Orbit
<b>RAAN</b>	Right Angle of Ascending Node
<b>SAGIN</b>	Space-Air-Ground Integrated Network
<b>SGP4</b>	Simple General Perturbations 4
<b>SOP</b>	Satellite Observation Point
<b>TEME</b>	True Equator Mean Equinox
<b>TLE</b>	Two-Line Element set
<b>WGS84</b>	World Geodetic System 84



---

## List of Figures

---

2.1	SOP-based transformation from ECEF to ENU coordinates (adapted from [4]) . . . . .	5
2.2	Transformations applied to coordinates of SGP4-based orbits in space_Veins . . . . .	5
2.3	Minimum elevation angle $\theta_{lim}$ for which satellite-ground communication is modeled to be possible. . . . .	7
2.4	Five of the Keplerian orbital elements: inclination $i$ , semi-major axis length $a$ , Right Angle of the Ascending Node (RAAN) $\Omega$ , Argument of Perigee / Periapsis (AOP) $\omega$ and true anomaly $f$ . . . . .	8
2.5	Definition of inclination via two lines perpendicular to the reference-orbital plane intersection line; in a plane perpendicular to reference plane and orbital plane . . . . .	9
2.6	Satellite position at moment $t$ and the position predicted by the mean motion $n$ at $t + 1$ , and their velocity vectors . . . . .	11
2.7	The circular orbital elements: inclination $i$ , radius $r$ , Right Angle of the Ascending Node (RAAN) $\Omega$ , circular anomaly $\Theta$ . . . . .	12
2.8	Spherical coordinates: Azimuth angle $\Phi$ , polar angle $\rho$ and radius $r$ for the point $P$ , and their relation to the Cartesian axes [23]. . . . .	13
2.9	Polar coordinates: Azimuth angle $\Phi$ and radius $r$ for the point $P$ , and their relation to the Cartesian axes [23]. . . . .	14
4.1	Transformations applied to coordinates of Keplerian orbits created in poliastro . . . . .	24

- 4.2 Visualization of the orbits of the IRIDIUM-100 satellite in the SGP4 and Keplerian model in the ECEF coordinate system ITRE, the first coordinate system of the transformation pipelines that is used for both models, simulated for 12 hours. In contrast to an ECI perspective, the orbits rotate with Earth. Even though at the same start time, the position is clearly different, the trajectories from there on are so similar on this scale that they seem to overlap, sometimes showing the Keplerian orbit, sometimes the SGP4 one - e.g. on the very left. Because in this instance, the simulation start wall-clock time is at a quite high distance to the epoch, there is a significant difference between the starting positions. . . . . 25
- 4.3 Movement on a circular orbit in 2D using spherical coordinates: The size of the orbit is determined by the radius  $r$ . Due to the alignment of ascending node direction and x-axis, the position at TLE epoch is given by the circular anomaly  $\Theta$ , from which the azimuth angle at simulation start  $\Phi_{2D}(t_0)$  is calculated by adding  $n(t_0 - t_{TLE}) \bmod 360^\circ$  using mean motion  $n$ . The position of the next simulation step is specified by the azimuth angle  $\Phi_{2D}(t_0 + \Delta t) = \Phi_{2D}(t_0) + n \cdot \Delta t \bmod 360^\circ$ . 28
- 4.4 In any plane in 3D through the TEME coordinate center, an arbitrarily oriented 2D coordinate system w.r.t. the center can be imagined by extending two perpendicular vectors  $a$  and  $b$  from the center to an imagined x- and y-axis, on which unit vectors specify unit lengths - for example the one consisting of the axes  $x_1$  and  $y_1$  using unit vectors  $a_1$  and  $b_1$ , or the one consisting of the axes  $x_2$  and  $y_2$  as another example. 29
- 4.5 Unnormalized vectors  $a$  and  $b$  when equatorial and orbital plane coincide, for an inclination of  $0^\circ$ . The angles are RAAN  $\Omega$  and azimuth angle of  $a$   $\Phi_a$ . . . . . 31
- 4.6 Unnormalized vectors  $a$  and  $b$  when the orbital plane is perpendicular to the equatorial plane, i.e. the inclination is  $90^\circ$  and  $b$ 's polar angle is  $0^\circ$ , while its azimuth angle can have any value. . . . . 31
- 4.7 Unnormalized vectors  $a$  and  $b$  when the inclination  $i \notin \{0^\circ, 90^\circ, 180^\circ\}$ . With a depicted inclination  $< 90^\circ$ ,  $b$ 's azimuth angle is  $\Phi_a + 90^\circ$ , and its polar angle  $\rho_b = 90^\circ - i$ . . . . . 32
- 4.8 Transformations applied to coordinates of circular orbits; fully in C++ 33
- 4.9 Initialization of a `CirclePlane` instance as implemented because of a previous approach that used two SGP4-derived points on the orbit that is to be modeled: unnormalized perpendicular  $a$  and  $b$  from their vectors from the coordinate system center in the orbital plane.  $b$  is calculated by applying the cross-product twice. . . . . 34

4.10 Visualization in the ECEF system ITRF of the orbits of the IRIDIUM-100 satellite, with an eccentricity of ca. 0.0002, in the SGP4 and Circular model, simulated for 12 hours. Even though at the same start time, the position is clearly different, the trajectories from there on are so similar on this scale that they seem to overlap, e.g. on the very left. In contrast to figure 4.2, the parts of the trajectories directly in the middle are visibly different. Because in this instance, the simulation start wall-clock time is at a quite high distance to the epoch, there is a significant difference between the starting positions.	36
5.1 Eccentricities of the 5 constellations considered in the simulation study	39
5.2 Inclinations of the 5 constellations considered in the simulation study	40
5.3 Average SGP4-TEME radii of the 5 constellations considered in the simulation study over a 12-hour period . . . . .	40
5.4 Mean motion and approximate velocity of the 5 constellations considered in the simulation study . . . . .	41
5.5 TLE epochs and epoch median for the 5 constellations. Three outliers have been removed for this visualization to achieve a greater resolution: One <i>Starlink</i> epoch on the 10th October, and two <i>SatNOGS</i> epochs on the 4th and 17th October. . . . .	42
5.6 Obstructions at 42.698156°N, 23.319892°E in Sofia from Google StreetView. The minimum elevation angle of 63.43° was estimated for obstructions like the building in the yellow box and ground positions like in the middle of the road half next to it. . . . .	48
5.7 Obstructions at 0.000127°N, 78.449996°W in Quito from Google StreetView. . . . .	49
5.8 Euclidian distances of the Keplerian and circular model to SGP4 of IRIDIUM 128 (subconstellation <i>IridiumNEXThigh</i> ) in ENU/OMNeT++ coordinates w.r.t. the Quito location . . . . .	56
5.9 Absolute differences of the Euclidian distances between the alternative models and SGP4 in ITRF and ENU/OMNeT++ . . . . .	57
5.10 Euclidian distances of the Keplerian and circular model to SGP4 of IRIDIUM 128 in ENU/OMNeT++ coordinates w.r.t. the Sofia location	58
5.11 Euclidian distances of the Keplerian and circular model to SGP4 of IRIDIUM 163 in ENU/OMNeT++ coordinates w.r.t. the Quito location	60
5.12 Euclidian distances of the Keplerian and circular model to SGP4 of ONEWEB 53 . . . . .	60
5.13 Euclidian distances of the Keplerian and circular model to SGP4 of ONEWEB 414 . . . . .	61

5.14 Euclidian distances of the Keplerian and circular model to SGP4 of IRIDIUM 107 in ENU/OMNeT++ coordinates w.r.t Quito . . . . .	62
5.15 Euclidian distances of the Keplerian and circular model to SGP4 of ONEWEB 17 . . . . .	62
5.16 IridiumNEXThigh ITRF z coordinates of the maxima and minima of the distances from the circular model to SGP4 . . . . .	65
5.17 OneWebHigh ITRF z coordinates of the maxima and minima of the distances from the circular model to SGP4 . . . . .	66
5.18 Euclidian distances of the Keplerian and circular model to SGP4 of POPACS 3 in ENU/OMNeT++ coordinates w.r.t Quito . . . . .	75
5.19 Euclidian distances of the Keplerian and circular model to SGP4 of PROSPERO(BLACKARROW) . . . . .	76
5.20 Elevation angles and availability periods resulting from it of the SatNOGS LAPAN-A2 satellite's orbit w.r.t. Quito comparing the Keplerian model to SGP4. Where the elevation angle is below the minimum elevation angle of $31.73^\circ$ , $0^\circ$ is plotted to clearly distinguish the positions of the availability periods in simulation time. . . . .	80
5.21 Zenith shifts of LAPAN-A2 in the Keplerian model w.r.t. Quito and SGP4	81
5.22 Zenith shifts of the ONEWEB 456 w.r.t. Quito . . . . .	81
5.23 Delay between shifted communication periods of the LAPAN-A2 satellite's orbit. Outside of the availability periods cannot be communicated, hence the delay is 0. . . . .	83
5.24 Delay between less shifted communication periods of the LAPAN-A2 satellite's orbit. Outside of the availability periods cannot be communicated, hence the delay is 0. . . . .	84

---

## List of Tables

---

5.1	Constellations, subconstellations, and their parameters median epoch, number of TLEs, inclination $i$ (rounded) range and range of altitudes averaged per orbit (rounded), which is derived from the orbits' average radii in TEME in a 12-hour SGP4 simulation and the WGS84 ellipsoid's mean radius of the three semiaxes. For SatNOGS, two inclination intervals are specified to clearly show how polar some of its orbits are. . . . .	44
5.2	Sums of between-model distances over the whole simulation period for the orbit of IRIDIUM 128. . . . .	59
5.3	Backward propagation simulation seconds of circular-to-SGP4 distance maxima and minima of IRIDIUM 163 . . . . .	63
5.4	Forward propagation simulation seconds of circular-to-SGP4 distance maxima and minima of IRIDIUM 163 . . . . .	63
5.5	Backward propagation simulation seconds of circular-to-SGP4 distance maxima and minima of ONEWEB 53 . . . . .	64
5.6	Forward propagation simulation seconds of circular-to-SGP4 distance maxima and minima of ONEWEB 53 . . . . .	64
5.7	Results of linear fitting the Circular-to-SGP4 distances in backward and forward propagation, where of the general function $y = a + b \cdot x$ the growth factor is $b$ and the epoch distance is $a$ . The average distances result from roughly 4.5 hours of backward and forward propagation. . . . .	67
5.8	Results of linear fitting the Keplerian-to-SGP4 distances in backward and forward propagation. . . . .	68

- 5.9 Circular classes of orbits whose  $\approx 4.5$  hours backward propagation average distance to SGP4 is larger than/smaller than their  $\approx 4.5$  hours forward propagation average distance to SGP4 of all constellations and subconstellations, the ratios of their orbit number to the total constellation/subconstellation orbit number, and the average distances of forward and backward of all orbits of each class. The yellow markings show for which classes in-class average distance comparison is not possible between the Keplerian and circular model because they don't occur in both. The other markings are in support of findings explained in the text. . . . . 70
- 5.10 Keplerian classes of orbits whose  $\approx 4.5$  hours backward propagation average distance to SGP4 is larger than/smaller than their  $\approx 4.5$  hours forward propagation average distance to SGP4 of all constellations and subconstellations, the ratios of their orbit number to the total constellation/subconstellation orbit number, and the average distances of forward and backward of all orbits of each class. The yellow markings show for which classes in-class average distance comparison is not possible between the Keplerian and circular model because they don't occur in both. The other markings are in support of findings explained in the text. . . . . 71
- 5.11 Worst-case distances of the circular and Keplerian model to SGP4 in 6 hours of backward and forward propagation, which are extrapolated to 24 hours of propagation each. Worst-case means that the highest distance reached between an orbit of SGP4 and the respective other model per (sub)constellation is displayed. The cells marked in yellow are the highest values of the constellation in the circular and Keplerian model respectively. The OneWebMiddle subconstellation did not contain any orbits which were propagated for 6 hours after their epoch. 73
- 5.12 Eccentric to-SGP4 distance metrics comparable to tables 5.9, 5.10 and 5.11 . . . . . 76

---

## Bibliography

---

- [1] P. Yue, J. An, J. Zhang, J. Ye, G. Pan, S. Wang, P. Xiao, and L. Hanzo, “Low Earth Orbit Satellite Security and Reliability: Issues, Solutions, and the Road Ahead,” *IEEE Communications Surveys & Tutorials*, vol. 25, no. 3, pp. 1604–1652, 2023.
- [2] O. Kordheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff, et al., “Satellite Communications in the New Space Era: A Survey and Future Challenges,” *IEEE Communications Surveys and Tutorials*, vol. 23, no. 1, pp. 70–109, 2021.
- [3] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, “Software Defined Space-Air-Ground Integrated Vehicular Networks: Challenges and Solutions,” *IEEE Communications Magazine*, vol. 55, no. 7, pp. 101–109, 2017.
- [4] M. Franke and C. Sommer, “Toward Space-Air-Ground Integrated Network Simulation with 4D Topologies,” in *19th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2024)*, Chamonix, France, Jan. 2024.
- [5] D. Vallado and P. Cefola, “Two-line element sets - Practice and use,” in *Proceedings of the 63rd International Astronautical Congress (IAC)*, vol. 7, Naples, Italy: International Astronautical Federation (IAF), Oct. 2012, pp. 5812–5825.
- [6] J. J. Lissauer and I. de Pater, *Fundamental Planetary Science: Physics, Chemistry and Habitability*. Cambridge University Press, 2019, pp. 25–30.
- [7] C. Wen, C. Zhang, Y. Cheng, and D. Qiao, “Low-Thrust Transfer Between Circular Orbits Using Natural Precession and Yaw Switch Steering,” *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 7, pp. 1371–1378, 2021.
- [8] M. Thammawichai and T. Luangwilai, “Data-driven satellite orbit prediction using two-line elements,” *Astronomy and Computing*, vol. 46, 2024.
- [9] N. Khairallah and Z. M. Kassas, “Ephemeris Tracking and Error Propagation Analysis of LEO Satellites with Application to Opportunistic Navigation,” *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–17, 2023.

- [10] X. Gao, Z. Li, Q. Chen, D. Ding, A. Peng, et al., “Prediction of Orbit Decay for Large-Scale Spacecraft considering Rarefied Aerodynamic Perturbation Effects,” *International Journal of Aerospace Engineering*, vol. 2022, 2022.
- [11] C. Sommer, R. German, and F. Dressler, “Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis,” *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, 2011. DOI: 10.1109/TMC.2010.133.
- [12] L. Mészáros, A. Varga, and M. Kirsche, “INET Framework,” in *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem*, A. Virdis and M. Kirsche, Eds. Springer International Publishing, 2019, pp. 55–106.
- [13] A. Varga and R. Hornig, “An overview of the OMNeT++ simulation environment,” in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, Marseille, France: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Mar. 2008.
- [14] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, “Microscopic Traffic Simulation using SUMO,” in *21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, Hawaii: IEEE, Nov. 2018, pp. 2575–2582.
- [15] D. Vallado, P. Crawford, R. Hujsak, and T. Kelso, “Revisiting Spacetrack Report #3,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Keystone, Colorado: American Institute of Aeronautics and Astronautics, 2006.
- [16] V. K. Srivastava, P. Mishra, and B. Ramakrishna, “Satellite ephemeris prediction for the Earth-orbiting satellites,” *Aerospace Systems*, vol. 4, no. 4, pp. 323–334, 2021.
- [17] T. Kelso, “Orbital Coordinate Systems, Part I,” *Satellite Times 09/10 1995*, vol. 2, no. 1, pp. 80–81,
- [18] B. Rhodes, Skyfield: High precision research-grade positions for planets and Earth satellites generator, Astrophysics Source Code Library, record ascl:1907.024, Jul. 2019.
- [19] F. C. C. (FCC), ““Order and Authorization 22-91(DA/FCC 22-91),” pp. 1–74, 2022.
- [20] D. Nelson, *The Penguin Dictionary of Mathematics*, Fourth edition. Penguin Books Limited, 2008.
- [21] R. H. Battin, *An introduction to the mathematics and methods of astrodynamics*. American Institute of Aeronautics and Astronautics, 1999, pp. 210–212.



- [22] S. Mathavaraj and R. Padhi, *Satellite Formation Flying: High Precision Guidance Using Optimal and Adaptive Control Techniques*. Springer Nature, 2021, p. 21.
- [23] L. Papula, *Mathematik für Ingenieure und Naturwissenschaftler Band 3 - Vektoranalysis, Wahrscheinlichkeitsrechnung, Mathematische Statistik, Fehler- und Ausgleichsrechnung*. Springer Vieweg, 2016.
- [24] N. Cheng, W. Quan, W. Shi, H. Wu, Q. Ye, H. Zhou, W. Zhuang, X. Shen, and B. Bai, "A Comprehensive Simulation Platform for Space-Air-Ground Integrated Network," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 178–185, 2020.
- [25] J. Puttonen, L. Sormunen, H. Martikainen, S. Rantanen, and J. Kurjenniemi, "A System Simulator for 5G Non-Terrestrial Network Evaluations," in *22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Pisa, Italy: IEEE, Jun. 2021, pp. 292–297.
- [26] J. A. Fraire, P. Madoery, M. A. Mesbah, O. Iova, and F. Valois, "Simulating LoRa-Based Direct-to-Satellite IoT Networks with FLoRaSaT," in *23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Belfast, United Kingdom: IEEE, 2022, pp. 464–470.
- [27] A. Freimann, M. Dierkes, T. Petermann, C. Liman, F. Kempf, and K. Schilling, "ESTNeT: a discrete event simulator for space-terrestrial networks," *CEAS Space Journal*, vol. 13, pp. 39–49, 2021.
- [28] A. Valentine and G. Parisi, "Developing and experimenting with LEO satellite constellations in OMNeT++," in *8th OMNeT++ Community Summit (OMNeT++ 2021)*, Virtual Conference: arXiv, Sep. 2021.
- [29] X. Wang, X. Chen, H. Ye, Y. Liu, and G. Zhang, "Cloud-based experimental platform for the space-ground integrated network," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–20, 2020.
- [30] D. Kassing Simonthand Bhattacharjee, A. B. Águas, J. E. Saethre, and A. Singla, "Exploring the "Internet from space" with Hypatia," in *IMC '20: Proceedings of the ACM Internet Measurement Conference*, Virtual Conference: Association for Computing Machinery (ACM), Oct. 2020, pp. 214–229.
- [31] T. Schubert, L. Wolf, and U. Kulau, "ns-3-leo: Evaluation Tool for Satellite Swarm Communication Protocols," *IEEE Access*, vol. 10, pp. 11 527–11 537, 2022.
- [32] T. Pfandzelter and D. Bermbach, "Celestial: Virtual Software System Testbeds for the LEO Edge," in *Proceedings of the 23rd ACM/IFIP International Middleware Conference*, Quebec, Canada: Association for Computing Machinery (ACM), Nov. 2022, pp. 69–81.

- [33] B. Kempton and A. Riedl, "Network Simulator for Large Low Earth Orbit Satellite Networks," in *ICC 2021 - IEEE International Conference on Communications*, Virtual Conference, 2021, pp. 1–6.
- [34] W. Jiang, Y. Zhan, X. Xiao, and G. Sha, "Network Simulators for Satellite-Terrestrial Integrated Networks: A Survey," *IEEE Access*, vol. 11, pp. 98 269–98 292, 2023.
- [35] Y. Zhang, B. Wang, B. Guo, Y. Yuan, T. Dong, J. Yin, K. Li, X. Guo, and S. Huang, "A Research on Integrated Space-Ground Information Network Simulation Platform Based on SDN," *Computer Networks*, vol. 188, 2021.
- [36] M. Liu, Y. Gui, J. Li, and H. Lu, "Large-Scale Small Satellite Network Simulator: Design and Evaluation," in *3rd International Conference on Hot Information-Centric Networking (HotICN)*, Hefei, China, 2020, pp. 194–199.
- [37] A. Petrosino, G. Sciddurlo, S. Martiradonna, D. Striccoli, G. Piro, and G. Boggia, "WIP: An Open-Source Tool for Evaluating System-Level Performance of NB-IoT Non-Terrestrial Networks," in *22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Pisa, Italy: IEEE, 2021, pp. 236–239.
- [38] D. Vallado and P. Crawford, "SGP4 orbit determination," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Aug. 2008, pp. 6770–6799.
- [39] J. J. Morales, J. Khalife, U. S. Cruz, and Z. M. Kassas, "Orbit Modeling for Simultaneous Tracking and Navigation using LEO Satellite Signals," in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, Miami, Florida: Institute of Navigation, 2019, pp. 2090–2099.
- [40] A. D. Biria, "Revisiting universal variables for robust, analytical orbit propagation under the Vinti potential," *The Journal of the Astronautical Sciences*, vol. 70, no. 5, pp. 29–66, 2023.
- [41] J. Kast, "Comparison of circular orbit and Fourier power series ephemeris representations for backup use by the upper atmosphere research satellite onboard computer," in *Flight Mechanics/Estimation Theory Symposium 1988*, Greenbelt, Maryland: NASA.
- [42] E. Doll, G. Mistretta, R. Hart, D. Oza, D. Bolvin, C. Cox, M. Nemesure, D. Niklewski, and M. Samii, "Improved solution accuracy for TDRSS-based TOPEX/Poseidon orbit determination," in *Flight Mechanics/Estimation Theory Symposium, 1994*, Greenbelt, Maryland: NASA, pp. 179–193.
- [43] J. R. Vetter, "Fifty years of orbit determination," *Johns Hopkins APL Technical Digest*, vol. 27, no. 3, pp. 239–252, 2007.

- [44] Y. Hao, S. Zhang, X. Zheng, and J. Qu, "Influence of atmospheric density and area-to-mass errors on orbit prediction accuracy," in *Journal of Physics: Conference Series - 2022 International Symposium on Aerospace Engineering and Systems (ISAES 2022)*, vol. 2252, Virtual Conference: IOP Publishing.
- [45] S. Baghel, D. Mohanty, A. Tamrakar, and T. Dewangan, "Accuracy Assessment of SGP4 and TLE Orbital Information," *Mathematical Statistician and Engineering Applications*, vol. 71, no. 4, pp. 9675–9680, 2022.
- [46] S. Kavitha, P. Mula, M. Kamat, S. Nirmala, and J. G. Manathara, "Extended Kalman filter-based precise orbit estimation of LEO satellites using GPS range measurements," *IFAC-PapersOnLine*, vol. 55, no. 1, pp. 235–240, 2022.
- [47] J. Haidar-Ahmad, N. Khairallah, and Z. M. Kassas, "A Hybrid Analytical-Machine Learning Approach for LEO Satellite Orbit Prediction," in *25th International Conference on Information Fusion (FUSION)*, Linköping, Sweden: IEEE, 2022, pp. 1–7.
- [48] C. A. Shuman, H. J. Zwally, B. E. Schutz, A. C. Brenner, J. P. DiMarzio, V. Suchdeo, and H. A. Fricker, "ICESat Antarctic elevation data: Preliminary precision and accuracy assessment," *Geophysical Research Letters*, vol. 33, no. 7, 2006.
- [49] P. F. Easthope, "Examination of SGP4 along-track errors for initially circular orbits," *IMA Journal of Applied Mathematics*, vol. 80, no. 2, pp. 554–568, 2015.
- [50] W. Dong and Z. Chang-yin, "An Accuracy Analysis of the SGP4/SDP4 Model," *Chinese Astronomy and Astrophysics*, vol. 34, no. 1, pp. 69–76, 2010.
- [51] T. Kelso, "Validation of SGP4 and IS-GPS-200D Against GPS Precision Ephemerides," in *17th AAS/AIAA Space Flight Mechanics Conference*, Sedona, Arizona: American Institute of Aeronautics and Astronautics, 2007.
- [52] R. Wang, J. Liu, and Q. Zhang, "Propagation errors analysis of TLE data," *Advances in Space Research*, vol. 43, no. 7, pp. 1065–1069, 2009.
- [53] M. E. Wesam, X. Zhang, Z. Lu, and W. Liao, "An orbit determination using SGP4 propagator and Doppler shifts for cubesats," *Transactions of Nanjing University of Aeronautics and Astronautics*, vol. 35, no. 3, pp. 472–482, 2018.
- [54] J. L. C. Rodríguez and J. M. Garrido, "poliastro: a Python library for interactive astrodynamics," in *Proceedings of the 21st Python in Science Conference (SciPy 2022)*, Austin, Texas: SciPy, Jul. 2022, pp. 136–146.

- [55] A. M. Price-Whelan, P. L. Lim, N. Earl, N. Starkman, L. Bradley, D. L. Shupe, A. A. Patil, L. Corrales, C. Brasseur, M. Nöthe, et al., “The Astropy Project: sustaining and growing a community-oriented open-source project and the latest major release (v5. 0) of the core package,” *The Astrophysical Journal*, vol. 935, no. 2, pp. 167–186, 2022.
- [56] L. Papula, *Mathematik für Ingenieure und Naturwissenschaftler Band 1 - Ein Lehr- und Arbeitsbuch für das Grundstudium*. Springer Vieweg, 2014.
- [57] F. Mölder, K. P. Jablonski, B. Letcher, M. B. Hall, C. H. Tomkins-Tinch, V. Sochat, J. Forster, S. Lee, S. O. Twardziok, A. Kanitz, et al., “Sustainable data analysis with Snakemake,” *F1000Research*, vol. 10, 2021.
- [58] V. Van Eylen, S. Albrecht, X. Huang, M. G. MacDonald, R. I. Dawson, M. X. Cai, D. Foreman-Mackey, M. S. Lundkvist, V. S. Aguirre, I. Snellen, et al., “The orbital eccentricity of small planet systems,” *The Astronomical Journal; IOP Publishing*, vol. 157, no. 2, 2019.
- [59] Z. Qu, G. Zhang, H. Cao, and J. Xie, “LEO satellite constellation for Internet of Things,” *IEEE Access*, vol. 5, pp. 18 391–18 401, 2017.
- [60] K. Croissant, D. White, X. C. Álvarez, V. Adamopoulos, C. Bassa, R. Boumghar, H. Brown, F. Damkalis, I. Daradimos, P. Dohmen, et al., “An Updated Overview of the Satellite Networked Open Ground Stations (SatNOGS) Project,” in *Proceedings of the Small Satellite Conference*, Logan, Utah, 2022.
- [61] Office of Geomatics, “Department of Defense World Geodetic System 1984: Its Definition and Relationships with Local Geodetic Systems,” National Geospatial-Intelligence Agency (NGA), Tech. Rep. NGA.STND.0036\_1.0.0\_-WGS84, Jul. 2014.
- [62] Office of Geomatics, “The Universal Grids and the Transverse Mercator and Polar Stereographic Map Projections,” National Geospatial-Intelligence Agency (NGA), Tech. Rep. NGA.SIG.0012\_2.0.0\_UTMUPS, 2014.
- [63] O. Montenbruck and E. Gill, *Satellite Orbits*. Springer, 2000.
- [64] G. M. Kurtzer, V. Sochat, and M. W. Bauer, “Singularity: Scientific containers for mobility of compute,” *PloS ONE*, vol. 12, no. 5, 2017.
- [65] J. Akeret, L. Gamper, A. Amara, and A. Refregier, “HOPE: A Python just-in-time compiler for astrophysical computations,” *Astronomy and Computing*, vol. 10, pp. 1–8, 2015.