Digital Twinning in Non-Terrestrial Networks

- Increasing the Operational Life of Satellite Batteries -

Master Thesis

Ainoa Salas

Aalborg University
Electronics and IT

This project has been developed using Python.

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Digital Twinning in Non-Terrestrial Networks: Increasing the Operational Life of Satellite Batteries

Theme:
Scientific Theme

Project Period:
Fall 2024 - Spring 2025

Project Group:
946

Participant(s):
Ainoa Salas

Supervisor(s):
Israel Leyva Mayorga
Beatriz Soret

Copies: 1

Page Numbers: 60

Date of Completion:
June 3, 2025

Abstract:

This project focuses on developing a Physical Twin simulator of a Geostationary Orbit satellite constellation and the corresponding Digital Twin model and service. The main goal is to use the Digital Twin to ensure long life expectancy of the satellite batteries.
Regarding the battery modeling in the Digital Twin, the report introduces and compares Discrete Time Markov Chains and Auto-regressive models. It is found that, in terms of accuracy, the Markov model performs better. For the service implementation, Reinforcement Learning is used for satellite resource allocation, ensuring battery stability, highly related to battery durability.
The results show how it is possible to use the Digital Twin to guarantee battery stability, fulfilling the initial design purpose of extending the battery life expectation.

Contents

Preface

Aalborg University, June 3, 2025

This work presents the outcome of my project on Digital Twinning in Non-Terrestrial Networks, carried out as part of my studies in Computer Engineering.

Ainoa Salas
<asalas23@student.aau.dk>

Abbreviations

A list of the abbreviations used in this report, sorted in alphabetical order:

3GPP	3rd Generation Partnership Project
ACF	Auto-Correlation Function
AI	Artifical Intelligence
AR	Auto-Regressive
CDF	Cumulative Density Function
CPS	Cyber-Physical System
C-Rate	Capacity-Rate
DS	Digital Shadow
DT	Digital Twin
DTMC	Discrete-Time Markov Chain
ESA	European Space Agency
FSL	Free-Space Loss
GEO	Geostationary Earth Orbit
GOES	Geostationary Operational Environmental Satellites
GS	Ground Station
IQR	Inter-Quartile Range
ITU	International Telecommunication Union
LEO	Low Earth Orbit
LoS	Line-of-Sight
MEO	Medium Earth Orbit
ML	Machine Learning
MSE	Mean Squared Error
NTN	Non-Terrestrial Network
PT	Physical Twin
Q1	Quartile 1
Q3	Quartile 3
QoS	Quality of Service
RL	Reinforcement Learning
SNR	Signal-to-Noise Ratio
SSS	Strict Sense Stationary
UML	Unified Modeling Language
WSS	Wide Sense Stationary

Chapter 1

Introduction

In recent years, there has been an increase in the usage and development of Digital Twins (DTs) [6]. This technology allows the user to have a virtual and faithful representation of a physical entity, called the Physical Twin (PT). One of the main advantages of Digital Twinning technology is its adaptability in different areas, mainly involving Cyber-Physical Systems (CPS): it can cover from a simple sensor system to a complex and detailed network infrastructure. DTs have also experienced a huge spreading due to the services they offer, which go far beyond monitoring or visualizing the status of the PT; innovative solutions can also be tested in the DT to optimize processes in the PT without compromising its normal operational tasks.

Building on the previous discussion, the main characteristic of a DT is a bi-directional data exchange with the PT [22]. The communications between a DT and PT can be seen as an automated closed loop where the the DT receives data and updates from the PT and provides a service directly to the physical entity. This duality in the data flow is what distinguishes a DT from a Digital Shadow (DS), where the communications can be modeled as an open loop. Therefore, a DS is a system able to only monitor and visualize the status of the physical entity and it is not going to be discussed in this project. Figure 1.1 illustrates the difference between DT and DS regarding the dual and single data flow, respectively.

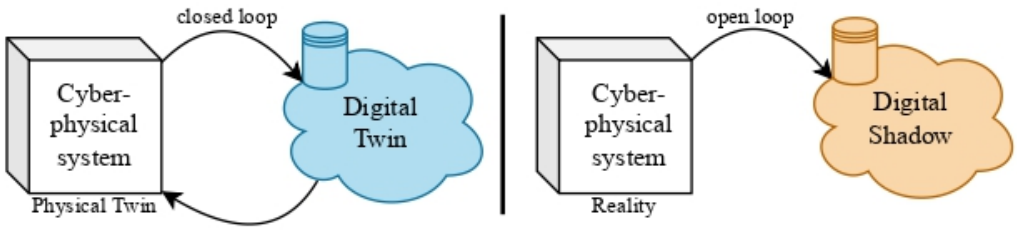


Figure 1.1: Communication schemes of Digital Twin and Digital Shadow towards the physical entity.

In the context of this project, a DT can be built to model a Non-Terrestrial Network (NTN) and deliver a wide of range of services. To clarify the concept, an example of an NTN is a satellite constellation orbiting around the Earth offering connectivity between users. The main advantages of using a NTN versus a Terrestrial one are the coverage offered even in rural areas, enhancing global communication, and the robustness against natural disasters [10]. However, there is an important consideration to take into account when working with NTNs, spacecrafts depend mainly on solar energy, which can not be always available due to orbital displacements. Besides that, the cost of launching a satellite is high, therefore it is crucial to try to increase the operational life of the satellite batteries. As introduced in [33], it is possible to optimize battery usage to slow down battery aging. For that reason, this project is based on using a DT to model and maximize the battery performance of the satellites of a NTN.

The aforementioned considerations can be rewritten to present the initial problem statement:
Can a Digital Twin of a satellite constellation be used to increase the operational life of the batteries while ensuring connectivity to the users?

Chapter 2

Technical analysis

To address the main question, how to extend satellite life using a DT, this chapter introduces the key theoretical concepts. Section 2.1 defines the satellite constellation, and Section 2.2 outlines the fundamentals of a DT. Section 2.3 covers satellite batteries, Section 2.4 discusses stochastic models, and Section 2.5 introduces Machine Learning (ML) techniques. Finally, Section 2.6 presents the evaluation metrics used in the project.

2.1 Satellite constellation definition

A satellite constellation can be defined as a collection of interconnected space stations orbiting around the Earth with a common objective. For example, in this project, the goal of the satellites is to offer connectivity to users. Even though most of the current Internet communications go through submarine cables, a modest percentage of data traffic is provided by NTN. These satellite networks can be extremely useful in isolated areas, where there is no wired infrastructure, or in case of emergencies, when natural disasters affect some of the communication elements.

As explained in [27], satellites are classified in function of their deployment altitude:

- Low Earth Orbit (LEO) satellites: their altitude is typically below 1000 km from the Earth. They are useful for taking aerial images due to its proximity to the surface.
- Medium Earth Orbit (MEO) satellites: they cover a wide range of altitudes, from LEO height up to the next category on this list. They are often used in navigation systems.
- Geostationary Earth Orbit (GEO) satellites: they are located 35.786 km above the Equator. Its main characteristic is that its relative position towards the Earth is fixed, providing reliable communications. They also offer a wide coverage area due its high position.

2.2 Digital Twin definition for satellite applications

As introduced in Chapter 1, a DT is a faithful representation of a PT that offers some added services to the users. Typically, the PT is a CPS, where computational and physical components interact in an environment. This project is based on the specific case of the DT of a NTN, defined as a constellation of satellites orbiting around the Earth communicating with Ground Stations (GSs) and users.

Figure 2.1 illustrates a possible system, composed of an interconnected PT and DT, that is used to clarify the concept of a DT. In this example, the PT is an eight-satellite constellation orbiting around the Earth. Satellite 1, *Sat*1, communicates directly with the GS, where the DT is stored and managed.

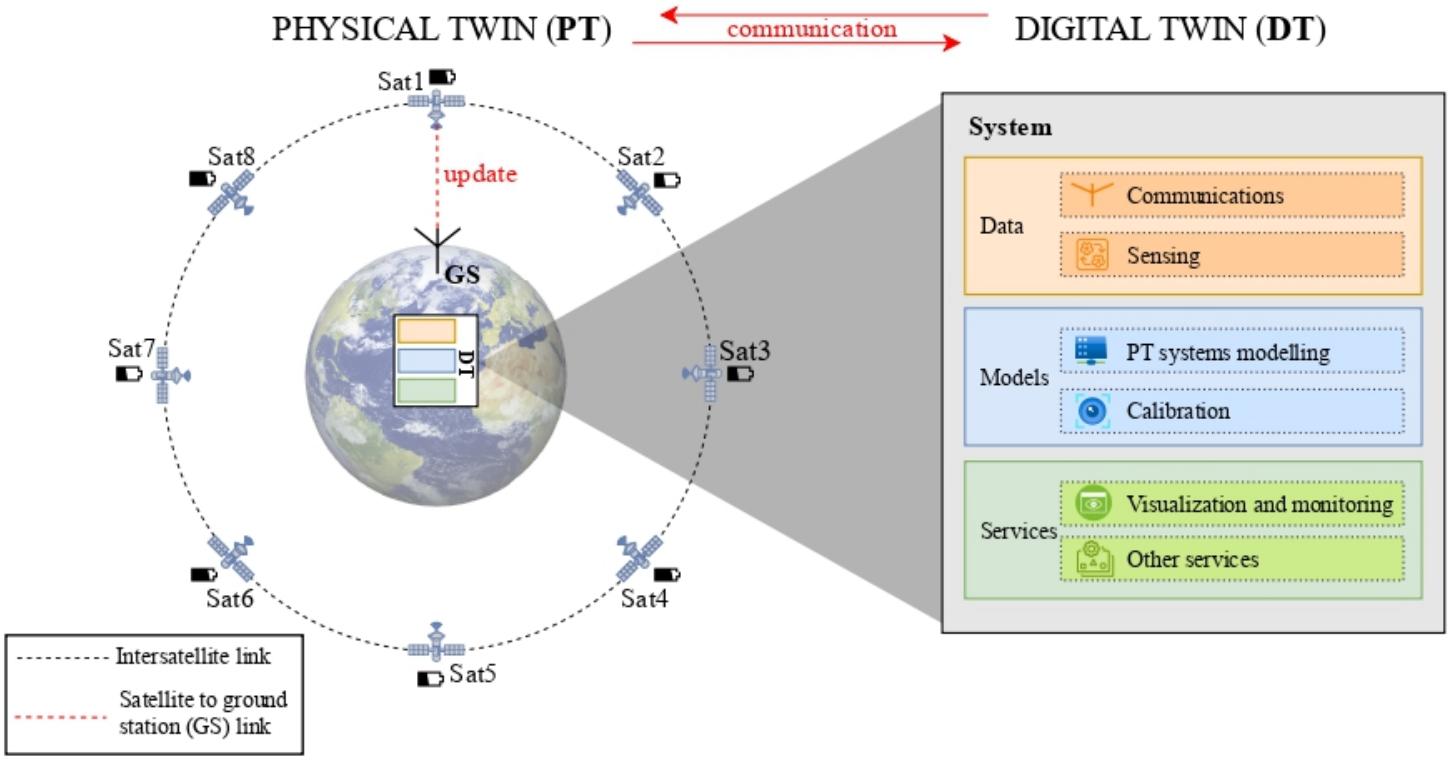


Figure 2.1: Digital Twin and Physical Twin system

Concerning the DT, its system can be subdivided into three subsystems:

- Data subsystem: it is responsible for receiving updates on the current state of the PT. This subsystem causes a drawback of using DT technology; the communications between the PT and the DT add an extra load to the normal activity of the PT.

- Models subsystem: as its name indicates, it contains the modeling of the PT systems i.e. the status behavior of system components. Using the updates received in the data subsystem, the models can be calibrated in order to guarantee fidelity with the PT.
- Services subsystem: it can offer the users of the DT different types of utilities. The most common are visualization and monitoring of the system. However, the main advantage of having a DT is the added services that can help improve the PT, such as status prediction using Artificial Intelligence (AI) applications.

From the description of its subsystems and as a summary, DT technology can be described in two words: fidelity and additionality. A DT is a faithful representation of a PT that, even if it adds additional communications, offers valuable services to the user.

2.3 Batteries: key elements, charging and discharging

Typically, satellites are fueled from a photovoltaic source, i.e. the Sun. However, the time when energy is needed may not match the catchment instant, leading to the need for a rechargeable battery storage system [7]. The characteristics and specifications of the battery depend on the satellite orbit altitude and its life requirements. In recent years, 98% of the satellites launched have used Lithium-Ion batteries [2]. These batteries are composed of cells, forming cell banks assembled in a series or parallel topology. Assuming there are N cell banks with an identical output voltage V_{bank}:

- If the cell banks are connected in series, the battery output voltage V is determined by adding all the individual voltages, Equation 2.1.

$$V = \sum_{i=0}^{N} V_{bank} = N \cdot V_{bank} \tag{2.1}$$

- If the cell banks are connected in parallel, the battery output voltage V is equivalent to the voltage of a single cell bank, Equation 2.2.

$$V = V_{bank} \tag{2.2}$$

Each battery has an associated maximum capacity C_{max} it can store, indicating the maximum energy it can supply if it is fully charged. During its lifetime, the battery is charged/discharged several times, so its current capacity fluctuates from 0 to C_{max}. The current capacity value C at time $t + \Delta t$ depends on the previous capacity value C at time t and its variation ΔC during the time difference Δt.

$$C_{t+\Delta t} = C_t + \Delta C \tag{2.3}$$

The variation ΔC can be related to the received/consumed power P using Ohm's law, as in Equation 2.4, where V corresponds to the battery voltage.

$$\Delta C = \frac{P}{V} \cdot \frac{\Delta t}{3600} \tag{2.4}$$

When the variation ΔC is a positive value, the battery is been charged. If the variation is negative, part of/all the energy stored in the battery has been consumed. Table 2.1 summarizes the units used for all the magnitudes in the equations above.

Magnitude	Symbol	Unit	Unit symbol
Voltage	V	Volts	V
Capacity	C	Ampere-hour	Ah
Power	P	Watts	W
Time difference	Δt	Seconds	s

Table 2.1: Summary of magnitudes and units

The main goal of this project is to increase the operational life of the batteries. Regarding that topic, [33] explains the relation between C-Rate (Capacity Rate) and battery degradation. C-Rate measures the speed at which a battery charges/discharges in relation to its maximum capacity. If the C-Rate is elevated, it affects the battery's long-term health. This concept is quite relevant through the whole project.

2.3.1 Power supplied

As mentioned at the beginning of the Section, solar panels charge the batteries from the Sun energy, using solar panels. The power a solar panel can obtain is described in Equation 2.5, where η is the panel efficiency, λ is the Sun irradiance[1] measured in W/m^2 and S is the panel surface measured in m^2 [32].

$$P = \eta \cdot \gamma \cdot S \cdot \cos\beta \tag{2.5}$$

The last part of Equation 2.5, β, refers to the angle between the sunlight and the normal of the solar panels. Equation 2.6 gives the minimum value of β when using single-axis solar tracking. It is extracted from [32].

$$\beta_m = \arccos\sqrt{1 - \cos^2\alpha \cdot \cos^2\theta} \tag{2.6}$$

- α is the angle between the orbital plane of the satellite and the sunlight.
- θ denotes the angle difference of the satellite current position from the greatest distance to the Sun.

  $\theta = \Delta t \cdot \omega = (t - t_0) \cdot \omega$ where t_0 denotes the time when the satellite is as far away as possible from the Sun and can not be greater than the orbital period.

[1] In the Earth and its corresponding man-made satellite orbits, it has a value of 1353 W/m^2.

For most satellites, there is a period when they do not receive sunlight because they are in an eclipse. This happens on satellites where Equation 2.7 is fulfilled, where R is the radius of the Earth and H, orbital height.

$$\alpha < \arcsin \frac{R}{R+H} \tag{2.7}$$

Figure 2.2 illustrates the effect of the eclipse: when satellites are orbiting in the grey cone, categorized as *Shadow* in the image, sunlight does not reach them[2]. The eclipse period can be defined as an angle interval of the orbit, $(-\phi_0, \phi_0)$. The center of the interval, zero, corresponds to the mid-part of the eclipse period, the furthest point of the orbit from the Sun. This interval is also represented in Figure 2.2 by the red part of the orbit and it can be computed following Equation 2.8.

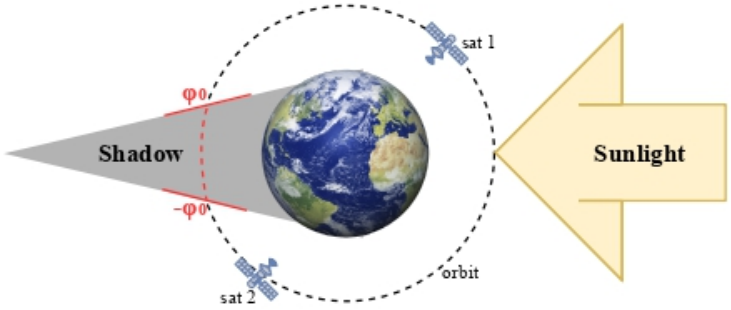


Figure 2.2: Eclipse illustration

$$\phi_0 = \arcsin \frac{\sqrt{R^2 \cos^2 \alpha - (2RH + H^2) \cdot \sin^2 \alpha}}{R+H} \tag{2.8}$$

Due to the eclipse effect, and considering that the point with greatest distance to the Sun corresponds to the middle of the eclipse, the power supplied by the Sun can be described with Equation 2.8.

$$P = \begin{cases} 0 & \text{if } |(t-t_0)\omega| < \phi_0 \\ \eta \cdot \gamma \cdot S \cdot \cos \beta_m & \text{otherwise} \end{cases} \tag{2.9}$$

[2]In the picture, the shadow is represented in 2D, however, in real life, it should be seen as a 3D cone.

2.3.2 Power consumed

The power consumed by a communications satellite can be divided into three main subsystems [16]: power required to transmit data to ground stations P_T, power consumed when processing received data from the users P_P, and power needed to keep the satellite in orbit P_0, following Newton's laws [3]. However, from now on, this project will not consider the two last items because it is uninterested in the communications field. In equation 2.10, P is the total power consumed.

$$P = P_T \tag{2.10}$$

First and foremost, the power used during data transmission corresponds to the required transmitted power by the satellite antenna to guarantee a certain Quality of Service (QoS) to the users located on Earth. This QoS depends on the desired rate of each user; depending on the type of service i.e. 1-2 Mbps approximately for browsing the Internet. The desired rate R can be related to the Singal-to-Noise power Ratio (SNR), and respective received power P_R and noise power P_N, using Shannon's capacity theorem [26], Equation 2.11. Shannon determines the maximum theoretical capacity of a communication channel in the presence of white noise.

$$R = B \cdot \log_2(1 + \text{SNR}) = B \cdot \log_2\left(1 + \frac{P_R}{P_N}\right) \tag{2.11}$$

The noise power P_N can be modeled as described in Equation 2.12, where k refers to the constant of Boltzmann, T the temperature of the system in Kelvins and B is the bandwidth of the transmission.

$$P_N = k \cdot T \cdot B \tag{2.12}$$

- The temperature of the system T is strictly related to the noise figure NF, equivalent to the noise factor F measured in decibels. The noise factor measures the SNR degradation suffered by the components of the equipment. Equation 2.13 corresponds to the noise factor, where T_0 is a standard of 290K.

$$F = 1 + \frac{T}{T_0} \tag{2.13}$$

- The bandwidth B value depends on the communication purpose, a different part of the spectrum is used for each objective. On its website [5], the European Space Agency (ESA) specifies the usage of the 1-40 GHz band for satellite/microwave communications. It also subdivides this band into smaller bands, each with a specific communications objective. Table 2.2 shows the subdivision of the spectrum. It is important to remark that higher frequency bands, which allow wider bandwidth, are more affected by rain, causing signal degradation.

Name	Spectrum band	Communications purpose
L-Band	1-2 GHz	Global Positioning System (GPS) carriers and satellite mobile phones.
S-Band	2-4 GHz	Weather radar, surface ship radar, and some communications satellites, especially of NASA.
C-Band	4-8 GHz	Satellite television, radio transmissions, and broadband communications.
X-Band	8-12 GHz	Military applications and radar systems.
Ku-Band	12-18 GHz	Satellite television broadcasting and high-speed internet.
Ka-Band	26-40 GHz	High-capacity satellite broadband and high-resolution radar.

Table 2.2: Spectrum bands for satellite communications

Finally, the power transmitted, P_T, can be related to the power received, P_R, using the Free Space Loss (FSL) propagation model [24]. In this model, a direct wave from the transmitter reaches the receiver via a line-of-sight (LoS) path without reflection. Equation 2.14 shows the above-mentioned relation, where G corresponds to the antenna gains, L the attenuation loss, λ the wavelength, and d the distance between transmitter and receiver.

$$P_R = \frac{P_T \cdot G}{L} = P_T \cdot G \cdot \left(\frac{\lambda}{4\pi d}\right)^2 \rightarrow P_T = \frac{P_R}{G} \cdot \left(\frac{4\pi d}{\lambda}\right)^2 = P_R \cdot \alpha \tag{2.14}$$

In the FSL propagation model, the total gain G corresponds to the gain of the transmitter multiplied by the gain of the receiver, Equation 2.15. For example, in the case of downlink from the satellite to the users, it would be the transmitter antenna gain of the satellite and the receiver antenna gain from the ground station.

$$G = G_T \cdot G_R \tag{2.15}$$

In order to compute the gain of the satellite, its antenna is modeled as a parabolic antenna, Equation 2.16, where d refers to the diameter of the parabolic reflector and e_a the aperture efficiency.

$$G = \left(\frac{\pi d}{\lambda}\right)^2 \cdot e_a \tag{2.16}$$

2.4 Stochastic modeling

This section introduce two different approaches for stochastic modeling, a mathematical model that incorporates randomness or uncertainty. The last subsection focuses on how to compare the accuracy of both methodologies.

2.4.1 Discrete-Time Markov Chain

From [21], Discrete-Time Markov Chain (DTMC) models a sequence of states S of a discrete-time random process $X_n, n = 0, 1, 2, \ldots$. The most relevant characteristic of a Markov Chain is the state dependence on the previous ones: the probability of transitioning to a state only depends on the current state. Equation 2.17 formally shows this definition: the probability of being in state j at time $n+1$ is only conditioned by the state at time n.

$$p_{ij} = P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, X_{n-2} = i_{n-2}, \ldots) = P(X_{n+1} = j | X_n = i) \quad (2.17)$$

The set of the transition probabilities of all the states determines the transition matrix P_T of the DTMC, where the sum of each row r needs to be 1, $\sum_{k=0}^{m} p_{rk} = 1 \ \forall r$. For a DTMC with m states, P_T is the following one:

$$P_T = \begin{pmatrix} p_{00} & p_{01} & \cdots & p_{0m} \\ \vdots & \ddots & \ddots & \vdots \\ p_{m0} & \cdots & \cdots & p_{mm} \end{pmatrix}$$

DTMC can also be represented graphically. For example, Figure 2.3 is a 5-state DTMC. The graphic representation shows the transition probabilities between states.

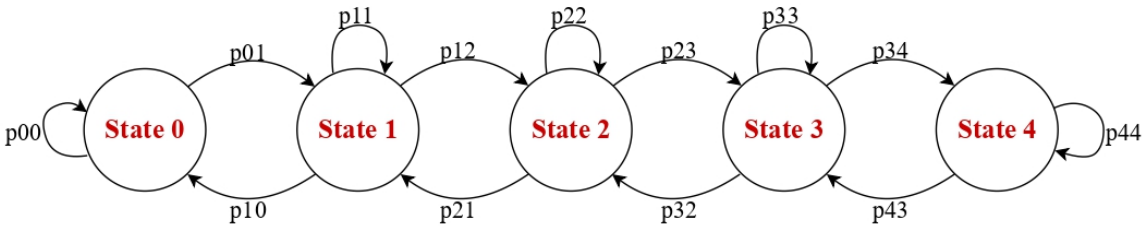


Figure 2.3: DTMC representation

The transmission matrix P_T of the Figure 2.3 is:

$$P_T = \begin{pmatrix} p_{00} & p_{01} & 0 & 0 & 0 \\ p_{10} & p_{11} & p_{12} & 0 & 0 \\ 0 & p_{21} & p_{22} & p_{23} & 0 \\ 0 & 0 & p_{32} & p_{33} & p_{34} \\ 0 & 0 & 0 & p_{43} & p_{44} \end{pmatrix}$$

2.4.2 Auto-Regressive model

An Auto-Regressive (AR) model $X[n]$ defines a system where the current value depends on past values [14], Equation 2.18. The vector $\Phi = (\Phi_1, ..., \Phi_p)$ corresponds to the coefficients of the model, where p is the order of the model (the number of past values used), and $Z[n]$ is white noise $N(0, \sigma^2)$.

$$X[n] = \sum_{i=1}^{p} \Phi_i X[n-i] + Z[n] \tag{2.18}$$

In order to fit a time-varying series into an AR model, it needs to be Strict-Sense Stationary (SSS) process [13]. However, it is not an easy task to verify if a series is SSS because the joint probability distribution needs to remain constant over time shifts. For that reason, the criteria for fitting into an AR model can be loosened to Wide-Sense Stationary (WSS) processes. The conditions for a series being WSS are easier to verify:

- The mean is constant over time.
- The Autocorrelation Function (ACF), that computes the covariance between two samples separated by k units, only depends on the time lag k.

  For example, for a process $X[n]$ with 3 samples, the ACF R_X would be computed following Equation 2.19, where $R[0]$, $R[1]$ and $R[2]$ are the corresponding covariances for samples separated by 0, 1 and 2 unis respectively. One of the most important properties of the ACF is the symmetry, $R[k] = R[-k]$. This feature can be seen in the example; $R[-2] = R[2]$ and $R[-1] = R[1]$.

$$R_X = \begin{pmatrix} R_{00} & R_{01} & R_{02} \\ R_{10} & R_{11} & R_{12} \\ R_{20} & R_{21} & R_{22} \end{pmatrix} = \begin{pmatrix} R[0] & R[1] & R[2] \\ R[-1] & R[0] & R[1] \\ R[-2] & R[-1] & R[0] \end{pmatrix} = \begin{pmatrix} R[0] & R[1] & R[2] \\ R[1] & R[0] & R[1] \\ R[2] & R[1] & R[0] \end{pmatrix} \tag{2.19}$$

The ACF can also be used to obtain the parameters of the AR model, Φ and σ^2, using Yule-Walker equations [4]. In Equation 2.20, $\delta[m]$ corresponds to the Kronecker Delta function, specified in Equation 2.21.

$$R[m] = \sum_{k=1}^{p} \Phi_k R[m-k] \; + \sigma^2 \cdot \delta[m] \tag{2.20}$$

$$\delta[m] = \begin{cases} 1, & m = 0 \\ 0, & m \neq 0 \end{cases} \tag{2.21}$$

Yule-Walker equations for $m > 0$ can be written in matrix form and solved for $\Phi_{(1,p)}$, Equation 2.22.

$$\begin{pmatrix} R[1] \\ R[2] \\ R[3] \\ \vdots \\ R[p] \end{pmatrix} = \begin{pmatrix} R[0] & R[-1] & R[-2] & \cdots \\ R[1] & R[0] & R[-1] & \cdots \\ R[2] & R[1] & R[0] & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ R[p-1] & R[p-2] & R[p-3] & \cdots \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \\ \vdots \\ \Phi_p \end{pmatrix} \tag{2.22}$$

To compute the value of σ^2, the remaining equation, for $m = 0$, can be solved using the values obtained before, Equation 2.23.

$$R[0] = \sum_{k=1}^{p} \Phi_k R[m-k] \; + \sigma^2 \tag{2.23}$$

If a process is Gaussian, it can be considered WSS without verifying the conditions. In that case, the ACF can be estimated from the process itself using autocorrelation and Yule-Walker can be applied in the results to obtain the parameters of the AR model. Equation 2.24 shows the estimated ACF of the process $X[n]$ with N samples.

$$\hat{R}[k] = \frac{1}{N} X[n] * X[-n] = \begin{cases} \frac{1}{N} \sum_{n=1}^{N-k} X[n]X[n+k] & k = 0, 1, ..., N-1 \\ \hat{R}[-k] & k = -N-1, ..., -1 \\ 0 & |k| \geq N \end{cases} \tag{2.24}$$

Differencing

As mentioned before, for a process to be WSS, the mean needs to be constant over time. However, most of the real live process have a periodicity that makes the statistical properties change over time. In order to remove the trends, differencing subtracts the current value from a previous one, Equation 2.25. The value of k is the differencing interval.

$$x'_t = x_t - x_{t-k} \tag{2.25}$$

2.4.3 Mean Squared Error

In the direction of comparing the accuracy of both methodologies, DTMC and AR, Mean Squared Error (MSE) can be used, Equation 2.26, where N is the total number of samples and y_i and $\hat{y}_i$ the real and estimated data respectively. Due to the squaring performed, MSE is highly suitable when it is important to penalize large errors.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2.26}$$

2.5 Machine Learning Techniques

ML is a field of AI that tries to learn from data to make predictions, identify patterns, or improve decision-making. It has two approaches, depending on the nature of the data:

- Supervised learning: the machine is given a set of inputs with its desired outputs, also known as labels. The goal is to establish a generalized relationship that links inputs to outputs.
- Unsupervised learning: the objective of the algorithm is to find the structure of the data without knowing the labels.

2.5.1 K-Means

K-means performs unsupervised classification on a set of points, defining a centroid for each one of the classes or clusters. The main disadvantage is that it does not work properly with unbalanced data. In consequence, a derived methodology is introduced, Weighted K-means [20].
In plain text, Weighted K-means consists of:

1. K random cluster centroids are initialized.
2. Each point p from the dataset is assigned to the cluster K with the closest centroid, using Euclidean distance.
3. The centroids are recalculated using the weighted mean w of the total points N of the corresponding cluster computed in 2, Equation 2.27.

$$Centroid\ position = \frac{\sum_{i=0}^{N} p[i] * w[i]}{\sum_{i=0}^{N} w[i]} \tag{2.27}$$

4. The process is repeated from 2 until the centroid convergence is achieved, meaning there are no large variations on the centroid positions between iterations.

Figure 2.4 shows graphically how K-means works, the only difference between K-means and Weighted K-means relies on the recalculation of the centroids, K-means just uses the mean of the points of the corresponding cluster. In other words, K-means is a general version of Weighted K-means where all the weights are the same.

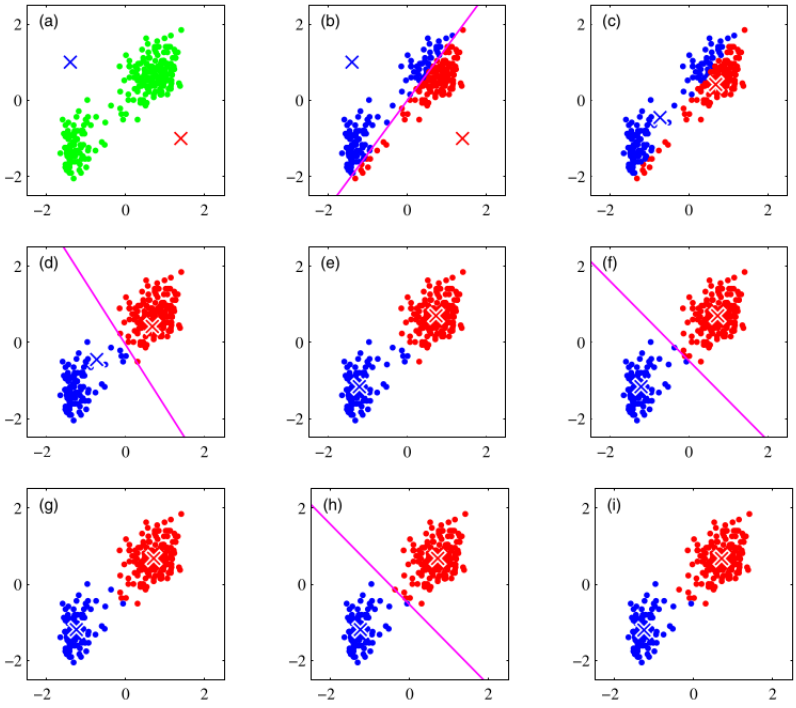


Figure 2.4: K-means illustration, extracted from [1]

2.5.2 Reinforcement Learning

From [30, 15], Reinforcement Learning (RL) is a ML approach situated between supervised and unsupervised learning. In a few words, RL can be defined as follows: there is an agent that decides between a determined number of actions in a dynamic environment to achieve a determined objective. In order to chose the most suitable action, the agent bases its decision on the expected reward. This reward can be described as the prize obtained for taking a certain action in a certain moment.

To delve deeper into RL, the main concepts and terms behind it are explained through a simple example, shown in Figure 2.5.

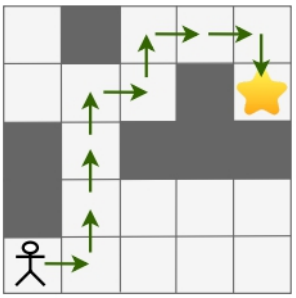


Figure 2.5: Reinforcement Learning example

The environment proposed in this case is a man, situated in the lower left corner of a $5x5$ grid, that has to reach a star. Here, the agent is the responsible of indicating the man the direction to take. The key principles of RL can be described and related to the example as follows:

- State space: set of possible situations of the environment. In this case, it corresponds to all the possible positions of the man in the grid. Therefore, all the cells except the dark gray ones.
- Action space: set of possible decisions the agent can take. In this case, the agent can decide to move the man up, down, left or right.

Q-Learning algorithm

In order to find the actions the agent needs to take to achieve its goal, there are many possible algorithms but one of the most common ones is Q-Learning [28]. As an introduction to this algorithm, the environment is defined as a finite discrete Markov process with n steps: the agent moves from one state S_n to another state S_{n+1} when taking certain action A_n and the expected reward R_n of taking the action A_n depends only on the current state.

The set of actions taken to reach the goal in an optimal way is called optimal policy. To find it, the agent needs to be trained. Q-Learning uses a Q-Table Q; each row of table corresponds to state S of the environment and each column, to an action A, creating pairs of action-state $Q(S, A)$. It is equivalent to the transition matrix of the Markov chain that defines the environment. The Q-table is initialized with zeros and updated during the training. For example, if the agent is in a certain state and a certain action would be beneficial for the final goal, the value of that cell, in the corresponding state row and action column, will end up being large.

Equation 2.28 defines the training procedure; $Q(S, A)$ is the value of the cell of the Q-Table corresponding to a pair of state-action, α is the learning rate, R is the reward obtained if action A is taken, γ is the discount factor and $\max Q(S', A')$ corresponds to the maximum estimated Q-Value of the next step.

$$Q(S, A) \leftarrow Q(S, A) + \alpha \left[R + \gamma \cdot \max Q(S', A') - Q(S, A) \right] \tag{2.28}$$

The learning rate α determines how much replacement of old information is caused by the new information, it ranges from 0 to 1 and the closer it is to 1, it means that new experience will update aggressively the Q-Table. The discount factor γ is used to determine how important are future rewards in comparison with current ones, keeping a balance between them. The reward R is a function that depends on the current state; it is chosen during the design of the algorithm and it is based on the environment and the goal of the RL agent. The most difficult part of the RL design is clearly the reward function design. For

example, in the previous case of a man trying to reach a star, a possible reward could be a function that gives higher values to the grid cells closer to the star that avoid getting the man stuck in dead ends.

The training of the agent is performed as it follows:

1. An initial state is decided.
2. An action is chosen. There are two ways of deciding the action:
   - Exploration: to try a random action.
   - Exploitation: to choose the action that will give the highest $Q(S, A)$ value possible.

   Normally, during the training, exploration and exploitation are combined, reducing the number of actions chosen by exploitation within the training time.
3. The Q-Table is updated using equation 2.28.
4. The procedure is repeated from step 2, taking as the new state the one given by doing the previous action.

2.6 Performance metrics

The initial problem statement from the Introduction revolved around increasing the operation life of the batteries while ensuring connectivity to the users. In that regard, two metrics for verifying the results obtained in this project are introduced here.

First of all, if a battery of one satellite drops til 0%, there is a complete depletion, the system is considered out of order, therefore it is not able to restart with its operational tasks. For that reason, one of the metrics chosen is the time until one of the batteries of the satellites is completely empty.

The second metric is related to ensuring connectivity to users. In order to measure the QoS offered by the satellite constellation, the amount of throughput not satisfied is quantified. To this effect, an observation regarding battery depletion is taken into account. Even though it is mentioned that the system would be out of order after dropping til 0%, in the computation of this metric, it is considered it can re-start, allowing to measure the throughput lost while the system was off.

2.7 Project scope and final problem statement

The scope of this project will address the design of a PT of a satellite constellation. The main focus will be addressed on the batteries. Following this, the corresponding DT will be also designed. This entity will be in charge of ensuring the life longevity of the batteries. Based on that and after revisiting all the theory introduced in this chapter, the initial problem statement can be reformulated in a more detailed way:
How can the battery status of satellites be accurately modeled with stochastic processes? How can battery models and AI algorithms be used within the Digital Twin scenario to extend the operational lifetime of the satellites?

This refined problem formulation highlights two key aspects: the need for accurate battery state modeling, and the application of these models within an intelligent decision-making framework. The model of the DT will represent the different battery status. In that case, there are two possible approaches: a DTMC or an AR Model. Both methodologies will be implemented and compared to determine which provides more accurate results. This addresses the first question of the problem statement.

After obtaining an accurate modeling of the system, a RL environment will be designed with the goal of extending the operational lifetime of the batteries. The algorithm will perform satellite resource allocation and will be part of the service offered by the DT. The design will ensure service coverage while monitoring battery status. The results obtained regarding the performance metrics after using the algorithm should answer the second question.

Chapter 3

System design

This chapter introduces the system model designed in this project. It is divided into two main sections: Section 3.1 focuses on explaining the design of the PT and Section 3.2 outlines the DT characteristics and methods.

3.1 Physical Twin

The PT of the satellites constellation is simulated in Python using parameters from the real world. The PT will include a constellation of four GEO satellites. A direct transmission from satellites to users is considered, not going through gateways. The following sections revolve around explaining the parameters of the PT:

- Firstly, Section 3.1.1 focuses on the characteristics of GEOs and their respective battery specifications.
- Section 3.1.2 describes how to place the satellites to guarantee a balanced load for each one.
- After that, Section 3.1.3 explains how to compute the rate required in each satellite based on the number of users in the Earth because, as explained in the previous chapter, the rate is directly related to the power consumed by the satellite. As a remark, the goal of the satellites will be to satisfy the rate required by the users.
- Finally, in Section 3.1.4, the communication parameters of the simulation of the PT are stated.

Section 3.1.5 shows how the structure of the Python code used to simulate the PT.

3.1.1 Characteristics of Geostationary Earth Orbit satellites

The rotational period of a GEO satellite is equal to approximately 24 hours, aligning with the rotational period of the Earth and implying that the relative position from the satellite to Earth is fixed. The altitude required to physically satisfy the previous condition is 35785 km. Given that GEO satellites are located on top of the Equator, users up to ± 81.3 degrees of latitude are included in the coverage ratio [12], that represents around one third of the Earth surface. Figure 3.1 illustrates the coverage area of a GEO satellite located in the Greenwich meridian (0 degrees of longitude).

The reasoning behind having a four GEO satellite constellation can be deducted at this point. With three satellites, it would already be possible to cover the whole Earth surface. However, one of the goals of this project is to perform resource allocation to keep the battery of all the satellites stable. Designing a PT with four satellites allows to obtain overlapping coverage areas, where it is necessary to perform a system management.

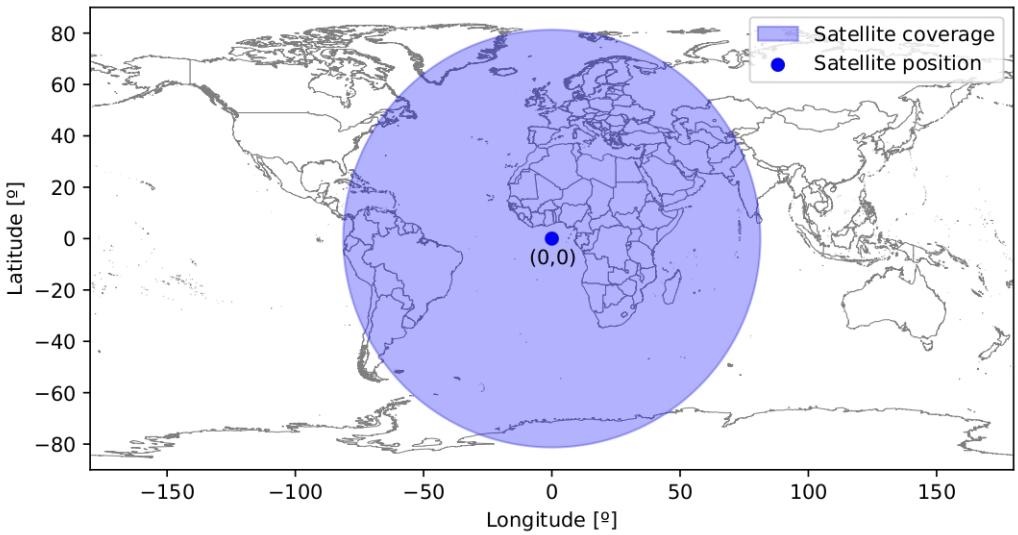


Figure 3.1: Example of coverage area of a GEO satellite

Regarding the battery specifications of each satellite, the parameters are extracted from [25] and summarized in Table 3.1. They correspond to the Geostationary Operational Environmental Satellites (GOES), built and launched in collaboration between NASA and National Oceanic and Atmospheric Administration (NOAA). Even if the goal of this satellite constellation is to provide continuous imagery and data on atmospheric conditions and solar activity and not to provide a certain rate to users, they are chosen as baseline because detailed information about their battery specifications is freely available. The parameters come from the satellites launched in November 2016, GOES-16.

Parameter	Value
Number of cell banks	12 connected in series
Voltage per cell bank	4 V
Number of batteries	2
Maximum capacity of a battery	140 Ah
Size of the solar panel	3.9624 m · 1.3716 m
Number of solar panels	5
Efficiency of solar panels	0.304

Table 3.1: Battery specifications and solar panel characteristics of GOES-16

Following Section 2.3, where the batteries key elements are introduced, and using the values from the previous table, the voltage in the batteries when the cell banks are connected in series is equal to 48 V and the total capacity, taking into account both batteries available, is 280 Ah in each satellite.

3.1.2 Satellite location: population clustering

The Earth population is not equally distributed throughout the world; there are highly populated areas, such as south Asia, and less dense zones, like the Pacific Ocean. Consequently, the location of a satellite, and the respective coverage area, needs to be consistent with the communication needs; zones with a high demand may benefit from having different satellites available at a time. Apart from that, in order to ensure a similar working time, in terms of battery life, the load given to each of the satellites should be similar.

As mentioned before, the main characteristic of a GEO satellite is that its relative position to the Earth is fixed. Given that fact and with prior knowledge of the Earth population distribution, it is possible to place the satellites assuring a similar load for each one. To achieve this, Weighted K-means clustering, introduced in 2.5, is used. In a satellite constellation scenario, the centroids represent the satellite locations and the data points, the location of the users. However, instead of representing each user with a single point, the Earth surface is divided in equal size zones, represented by a single point with a corresponding weight, depending on the area density.

The most straight-forward application of the algorithm would consist on using longitude and latitude as the coordinates x and y of the data points. Nevertheless, the Earth is a sphere and with that approach, the points on the different map-edges of the 2D representation would never be in the same cluster, generating non-optimal results in the satellite location. On that ground, Spherical K-Means [8] offers a solution. Each point is represented by a 3 dimensions (3D) vector and instead of using Euclidean distance when assigning a label, the angle difference between the point vector and the centroid vector is computed.

Algorithm 1 contains the pseudo-code for Spherical Weighted K-means methodology used.

Algorithm 1 Spherical Weighted K-means

```
dataset = user locations and corresponding weight
clusters = number of satellites
centroid[0:clusters] ← random initialization

while not convergence do
    for point in dataset do                    ▷ ** Label assignation to dataset points **
        angles ← Angle with centroids
        label ← index of argmin(angles)
    end for

    for i = 1 to len(centroid) do              ▷ ** Centroid recomputation **
        newCentroid[i] ← weighted mean of points assigned to i
    end for

    if newCentroid == centroid then            ▷ ** Convergence check**
        convergence ← True
    else
        centroid ← newCentroid
    end if
end while
```

3.1.3 Number of active users

Most of the current Internet infrastructure uses underground cables, however, 1% of the Internet traffic goes through satellites [18]. For that reason, the rate required from a determined satellite corresponds to the summation of the rate of computing the 1% of the number of active users within the coverage area of the satellite.
The following procedure obtains these results:

1. Natural Earth Data [19] proportionates a map delimitating each country of the Earth. This map also includes information about the population of each country. In order to obtain the number of users that may be connected to a satellite, it is combined with data from an International Telecommunication Union (ITU) map [9], where the percentage of users with Internet access at home per country is available. This percentage is used as a reference on how many citizens per country have access to a

laptop or a phone to establish a connection with a satellite. Both maps are attached in Appendix A.

2. The complete surface of Earth is divided into squared regions of 15º of latitude per 15º of longitude. For each region, the number of users with Internet access, based on the results of the previous step, is computed. A small example can be seen in Figure 3.2, where each number represents the millions of users of each cell when splitting a world region. The results of the whole world are available in Appendix A.

3. The last step consists of multiplying the previous results by 1%, obtaining only satellite Internet traffic, and adding the cells within the coverage area of the satellite.

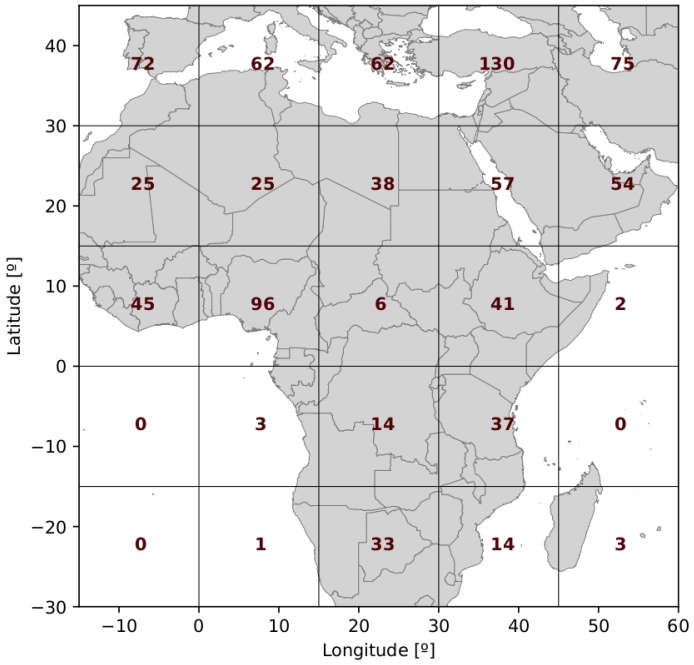


Figure 3.2: Number of users per cell in Africa, Middle East, and Southern Europe

The number of active users not only depends on the location but also on the day time [29, 11]. The traffic of each cell needs to be multiplied by a weighting factor depending on the hour of the day. This weighting factor is chosen from the interval $(0, 1)$. Figure 3.3 illustrates the factor per hour.

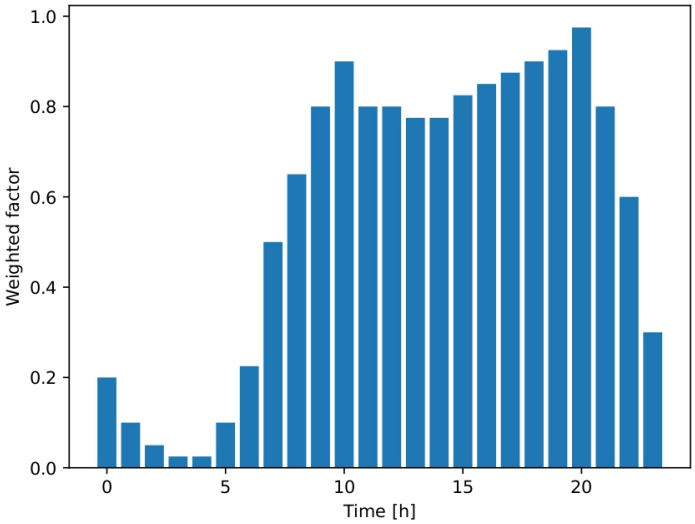


Figure 3.3: Weighted time, extracted from [11]

In order to get a more realistic number of users, an statistical parameter is introduced. If for every 24-hour simulation of the PT, the rate were computed using just a deterministic parameter, the battery model on the DT would be quite easy to calibrate. However, these values would not be faithful to the real human behavior. For that reason, the number of users obtained using the previous procedure is multiplied by a value x extracted from a Gaussian distribution. The chosen one is a Normal distribution with mean μ equal to 1 and standard deviation σ equal to 0.01. The value of the standard deviation is not large in order to avoid substantial discrepancies between simulations.

3.1.4 Communications system

Firstly, using the number of users obtained before, the rate required by the satellite to the users is computed. It is estimated that each user wants a connection of 1 Mbps, equivalent to browsing the Internet. This value is multiplied by the number of users within the coverage area and the final rate for a time instant is determined. As introduced in the previous chapter, the rate will give the value of the power consumed.

Regarding other communication parameters used for the PT, they are extracted from 3rd Generation Partnership Project (3GPP) standards [23]. It is important to remark than the connection considered is always down-link, meaning from the satellite to the user. From that specifications, the satellite antenna aperture is set as 22 meters and the spectral efficiency as 0.55. Using the formula introduced in Section 2.3.2, it is possible to calculate the transmission gain, Equation 3.1.

$$G_T = \left(\frac{\pi d}{\lambda}\right) \cdot e_a = \left(\frac{\pi \cdot 22}{\lambda}\right) \cdot 0.55 \tag{3.1}$$

The missing λ on the previous equation corresponds to the wave length and it can be deducted from the spectrum band used by the satellite communications. Following Table 2.2, from Section 2.3.2, where the different spectrum bands are introduced, the chosen one is L-Band (1-2 GHz) because it can be used for satellite mobile phones, individual users on Earth, and it is less affected by rain than higher frequency spectrum. In this case, the attenuation effects caused by the meteorology can be disregarded in the PT simulation. Delving into the spectrum, a satellite communication does not use the whole band for transmission; it depends on the purpose of the communications. In this case, the connection established is down-link, or space-to-Earth, so the sub-spectrum chosen is B_L 1.559-1.545 GHz. Eventually, the carrier frequency f_c of the transmission is the middle value of the bandwidth B_L, 1.552 GHz, and the wave-length λ can deducted using the electromagnetic wave relation of Equation 3.2, where c corresponds to speed of light.

$$\lambda = \frac{c}{f_c} = \frac{3 \cdot 10^8}{1.552 \cdot 10^9} = 0.1932\ m \tag{3.2}$$

As mentioned before, the PT simulation is done taking into consideration a direct connection between the satellite and the user. In that case, the reception gain G_R is equal to 0 dB. Due to that, the final gain from the system is equivalent to the transmission gain,

$$G = G_T \cdot G_R = G_T \cdot 1 = G_T \tag{3.3}$$

Finally, the temperature of the system is set as 290 Kelvins. This value is also extracted from the 3GPP standards and it corresponds to the temperature of a user on the Earth, because that is the final destination of the down-link communications.

3.1.5 Code structure for Physical Twin simulation

The PT is simulated in Python following the Unified Modeling Language (UML) diagram from Figure 3.4. There are three main classes, Constellation, Satellite and Battery, that contain the main attributes of a satellite constellation. The *Simulation Runner* is the controller in charge of rotating the constellation and tracking the battery status.

In a few words, the UML diagram has the following behavior: the *Simulation Runner* creates a constellation with a certain number of satellites and it rotates them while updating the respective battery status, that are computed according the rate needed by the users and the power received by the Sun.

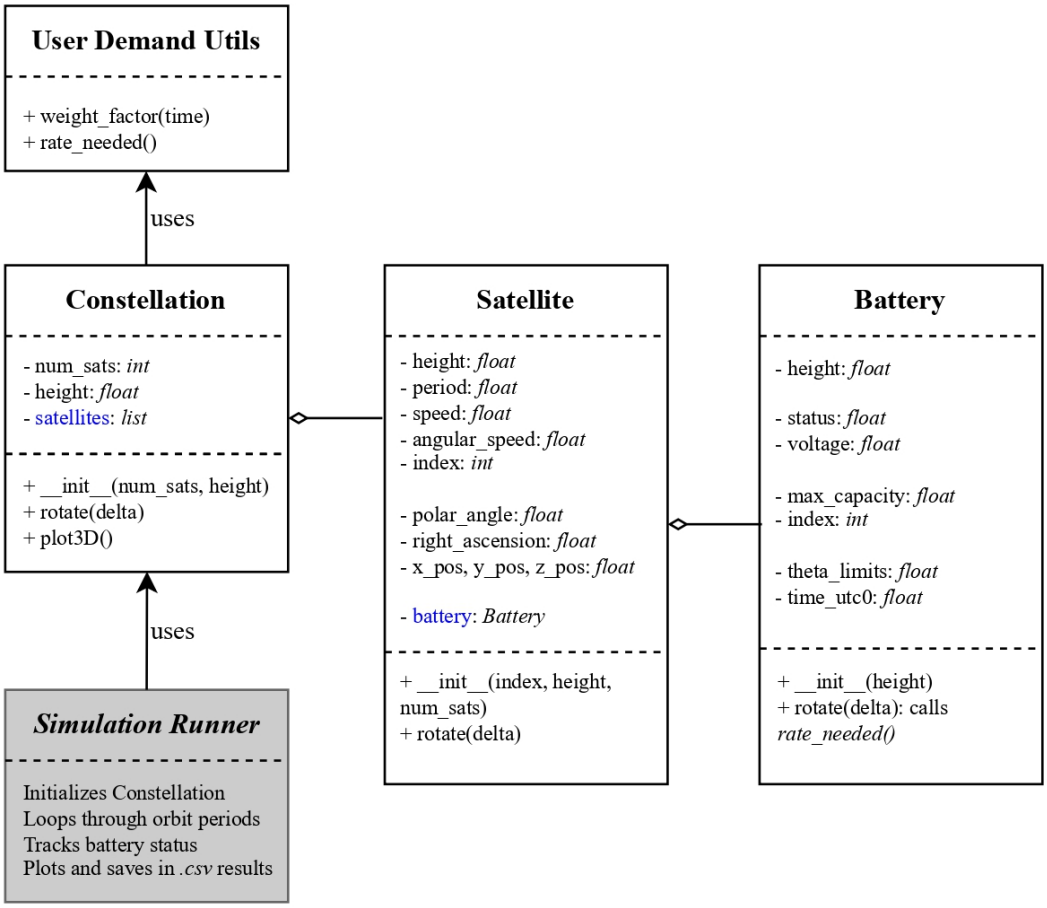


Figure 3.4: UML diagram of Physical Twin Simulator

The sampling period chosen to update the battery status is 15 minutes. There are two main reasons behind this number:

- It allows to obtain four different samples from every hour.
- It allows to run the simulation for an entire day in a feasible time.

3.2 Digital Twin

This section explains the main characteristics of the DT: the modeling used and the service offered. The DT will model the battery status of the satellites to, as part as its service, ensure a long life expectancy. An important detail to be remarked on from the beginning is the placement where the DT is stored; it is kept in its entirety on a GS, meaning the constellation will have direct communication to it. The reasoning behind this idea is based on simplicity: if the DT were stored in the satellites, it would probably need to be fractionated due to memory constraints and the communication between DT and PT would be more complex.

3.2.1 Modeling

In Section 2.4, two stochastic models were introduced, DTMC and AR models. Both of them can be used to model the battery status of the satellites in the DT and the one with best performance regarding accuracy will be chosen. As an initial remark, the sampling period of the models is 15 minutes, in order to be equivalent to the simulation period of the PT.

Discrete Time Markov Chain

Markov chains can be used to model the battery status of each one of the satellites in the DT, each one of the states represents a percentage of battery of the satellite. One of the advantages of this mathematical framework is its simplicity. Nevertheless, it can also become a drawback: it may not be able to cover all the details of the system. In the case of this report, the probability of changing from one state to another is highly dependent of the daytime period, e.g., the power consumption increases on peak hours and the power received decreases during eclipses. For that reason, this Subsection aims to propose a suitable time dependent Markov Chain model to reproduce the battery fluctuation in the DT as accurately as possible.

The approach chosen consists of splitting the orbital period and computing a Markov Chain for each one of the divisions. It is significant to note that if the number of Markov Chains is large, the complexity and storage can rapidly increase. Consequently, the total of splits needs to be done according to feasible computations capabilities. Given the fact of 24 hours rotation period, the most logical option is to compute a Markov Chain for each one of the hours. Equation 3.4 calculates the total number of Markov Chains stored in the DT, given the number of satellites of the constellation n_{sat}.

$$\#\text{ of Markov Chains} = n_{sat} \cdot 24 \tag{3.4}$$

Once the number of Markov Chains is clear, it is time to analyze in more detail the characteristics of each one, in particular, the number of states. As mentioned in the beginning,

each state represents a percentage of remaining battery of the satellite. For simplicity, it is established that the different Markov Chains per hour will have the same number of states. In order to guarantee a faithful representation of the constellation, different number of states will be tried and the accuracy compared using the MSE. The possible number of states for each Markov chain will be chosen from: 5, 11, 21, 41, 51 and 101. These numbers are decided on basis of getting to know if it is possible to have an accurate model with a few number of states and if there is a large difference between designs with many states.

Regarding the probability p_{ii} of being in the same state after a time step, it can be mathematically expressed as in Equation 3.5. The initial battery capacity is C_t, the capacity after a time step is C_{t+1}, k is the minimum difference between states that the Markov Chain can represent and C_{max} the maximum capacity of the battery.

$$p_{ii} = P\left(|C_t - C_{t+1}| < \frac{k}{100} \cdot C_{max}\right) \tag{3.5}$$

In order to get a better understanding of k, an example with a 5-state Markov Chain is explained:

- Having five states would be equivalent to have the following status of the battery: 0%, 25%, 50%, 75% and 100%. In that case, k would be equal to 25, meaning that if it is in state 0%, it would be there until there is a change of battery larger than 12.5%.
- Figure 3.5 helps to understand the discretization process. The battery percentage values obtained from the PT simulation are continuous, however, when modeling with a Markov Chain, the number of states is finite and not all the possible values can be represented.

  It is important to remark that all the states take an interval $\left(s - \frac{k}{2}, s + \frac{k}{2}\right)$, where s is the battery value represented by the corresponding state. Nevertheless, there is an exception in the first and last state, that represent $\left(0, s + \frac{k}{2}\right)$ and $\left(s - \frac{k}{2}, 100\right)$ respectively. This observation also affects the computation probabilities of this two states, Equation 3.6.

$$p'_{ii} = P\left(|C_t - C_{t+1}| < \frac{1}{2}\frac{k}{100} \cdot C_{max}\right) \tag{3.6}$$

The probability p_{ij} of being in a different state after a time step is quite similar to the probability p_{ii}. The main distinction is that the difference between the battery capacities needs to be greater than the difference between respective the states S_i and S_j.

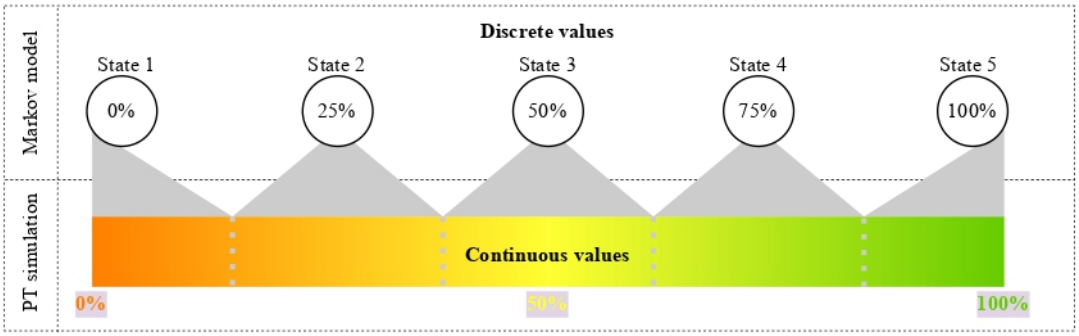


Figure 3.5: Markov Chain discretizations

Auto-regressive model

In order to model the battery status with an AR model, there is an important initial question to take into account: if it is necessary to difference the simulation values used for the prediction. First of all, the simulation of the batteries is highly dependent on time. This means that it will have a periodicity. In order to remove the periodicity, it is clearly necessary to difference, as detailed in Section 2.4. The differencing period should be equivalent to the Earth rotation.

Regarding the computation of the AR model, there are two implementations that will be tested: Full Yule-Walker and Sparse Yule-Walker. The first one consists of using all the lags available from the original data and, the second one uses only the dominant peaks of the Auto-Correlation Function (ACF). The reasoning behind this idea is to avoid obtaining noise dependent results. The simulation obtained by the PT is affected by randomness due to the Gaussian distribution used to calculate the number of users. In this case, the objective of the AR model is to predict the future battery states without taking into account the noise from the previous samples and, that means that Sparse Yule-Walker can have a better performance than Full Yule-Walker.

3.2.2 Service

The main goal of the DT is to increase the operational life of the batteries. In that regard, the objective is to avoid batteries being completely empty and fast charge/discharge, equivalent to high C-Rate. The design proposed for the DT is a RL algorithm able to fulfill that goals. Concerning the characteristics of the RL environment, the action- and state-space can be defined as follows:

- State-space: discretizations of the battery percentages of the satellites.

- Action-space: set of all possible ways the satellites can be assigned to cover the different Earth regions. The system will decide which satellite needs to be activated to cover specific Eart zones, based on the battery status. This process is called resource allocation.

The reward of the RL needs to reflect the main goal. For that reason, this are the design check-points: high penalization from drop-off below 0 % and higher rewards are given when the battery levels remain stable between time steps. The first checkpoint corresponds to avoiding battery depletion, and the second one to ensuring a low C-Rate. As explained in the previous chapter, a low C-Rate is a key parameter for preventing battery degradation.

The final design proposed of the service consists of the PT sending updates every hour to the DT regarding the battery status, and the Digital Twin replying with the actions for the next hour concerning resource allocation. Based on the current status of the batteries, the DT will use one of the models proposed before to predict the future status and choose the most suitable actions. Figure 3.6 shows a diagram of the complete service offered by the DT

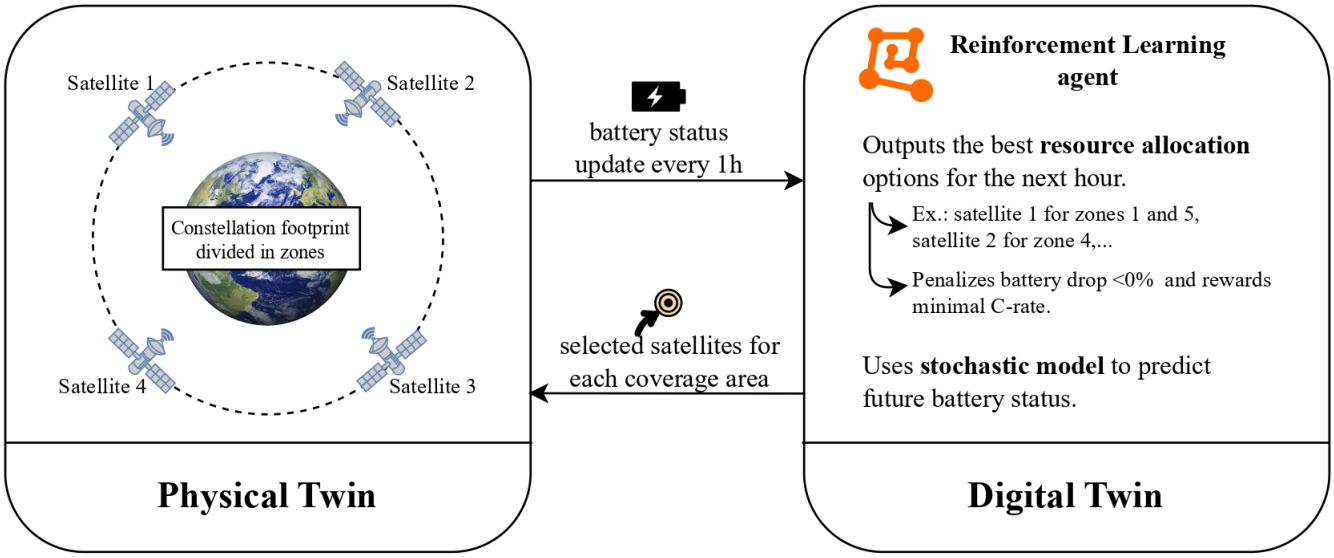


Figure 3.6: Diagram of the service offered by the DT

Chapter 4

Testing and results

As in the previous chapter, this one is also divided into PT and DT sections. Section 4.1 (PT) presents the main environmental results from the Physical Twin design, while Section 4.2 (DT) focuses on obtaining and comparing battery models, as well as developing and analyzing the service.

4.1 Physical Twin

The satellites of a constellation are in constant movement and, as detailed in Section 2.3.1, the amount of power supplied by the sun varies depending on the orbital position. Figure 4.1 illustrates the power supplied from the Sun in a satellite battery during an entire orbital period. The initial position of this satellite is the orbital point closest to the Sun.

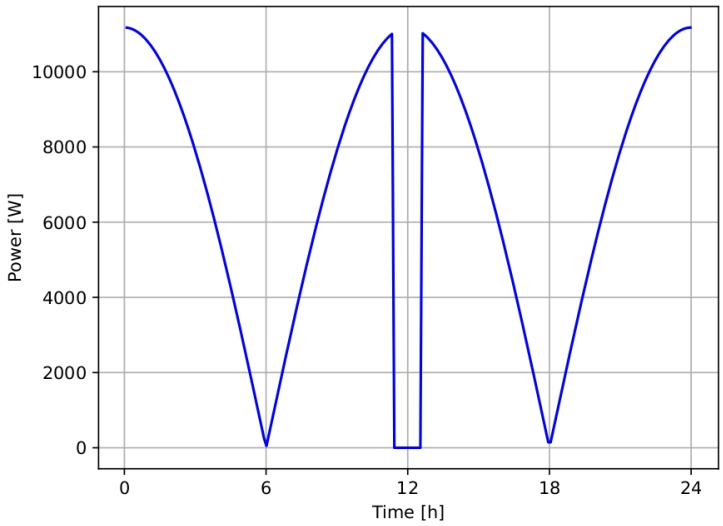


Figure 4.1: Power proportioned by solar energy

The variation of the power received depends on the angle of the solar panels with the Sun light and consequently, the orbital period time. In ideal conditions, when the solar panels are completely perpendicular to the Sun light, at the beginning of the period, the maximum power supplied is equal to approximately 11kW.

In the graph, it is also possible to notice the eclipse effect; when the satellite is located at half of the period, there is a time lapse where there is no power received. The time lapse is around one hour. Finally, when the satellite is at $\frac{1}{4}$ or $\frac{3}{4}$ of the orbital period, there is also no power supplied. This is due to the single axis solar track design of the solar panels, that does not offer any chance to the solar panels to catch Sun light if they are parallel to it.

4.1.1 Satellite location

As detailed in Section 3.1.2, Spherical Weighted K-means can be used to determine the position of the satellites. However, in the algorithm, the centroids have complete freedom, meaning they can be at any point of the space. This is not the case of GEO satellites, where the satellites relative position to the Earth is fixed to the Equator (zero degrees of latitude). If the previous condition is not considered, the algorithm gives consistent results on the centroids. Figure 4.2 shows the centroids obtained after running the algorithm one hundred times; it gives a similar location for all the centroids in the multiple realizations.

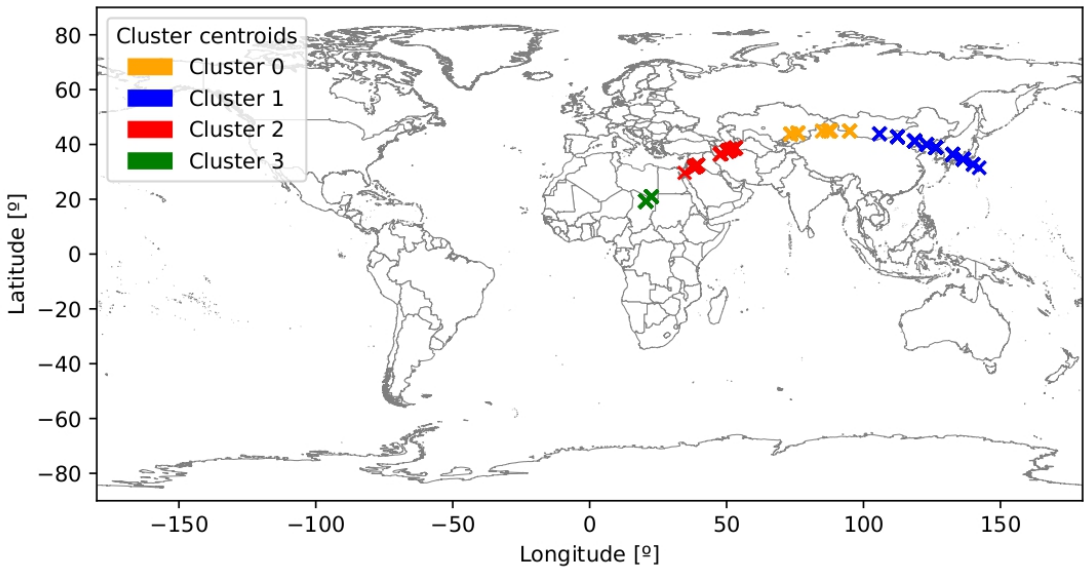


Figure 4.2: Multiple realizations of Spherical Weighted K-means without Equator restriction on the satellite position

Nevertheless, when the latitude of the centroids needs to remain fixed, the algorithm does not give consistent results among realizations. It is observed that the results have a strong dependency with the initialization of the centroids. For that reason, instead of starting with random positions, and choosing the results obtained, multiple realizations are executed to discover the most repeated outcomes. Figure 4.3 shows the results obtained after running the Spherical Weighted K-means algorithm a thousand times with random initializations. In the graph, it is possible to see how most of the outcomes converge in four peaks, corresponding to the satellite positions in relation to the Earth longitude. The final results obtained with the algorithm, locate the satellites in the longitudes of -87, 14, 87 and 114.

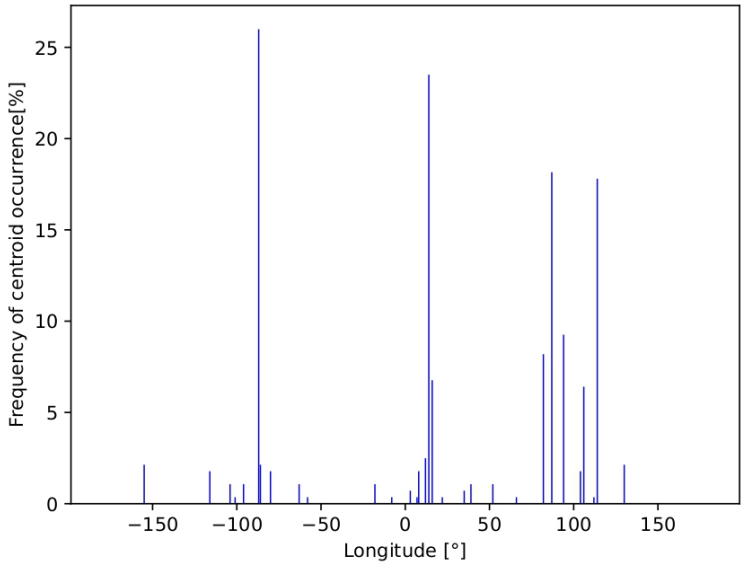


Figure 4.3: Frequency of centroid occurrence depending on longitude

4.2 Digital Twin environment

The main objective of the DT is to ensure a long life expectancy of the batteries. In order to achieve this, it is necessary to obtain a battery modeling that can be used as part of the DT service. Firstly, this section analyzes the performance of the models of the batteries. Secondly, it develops the proposed RL algorithm and it checks its efficacy.

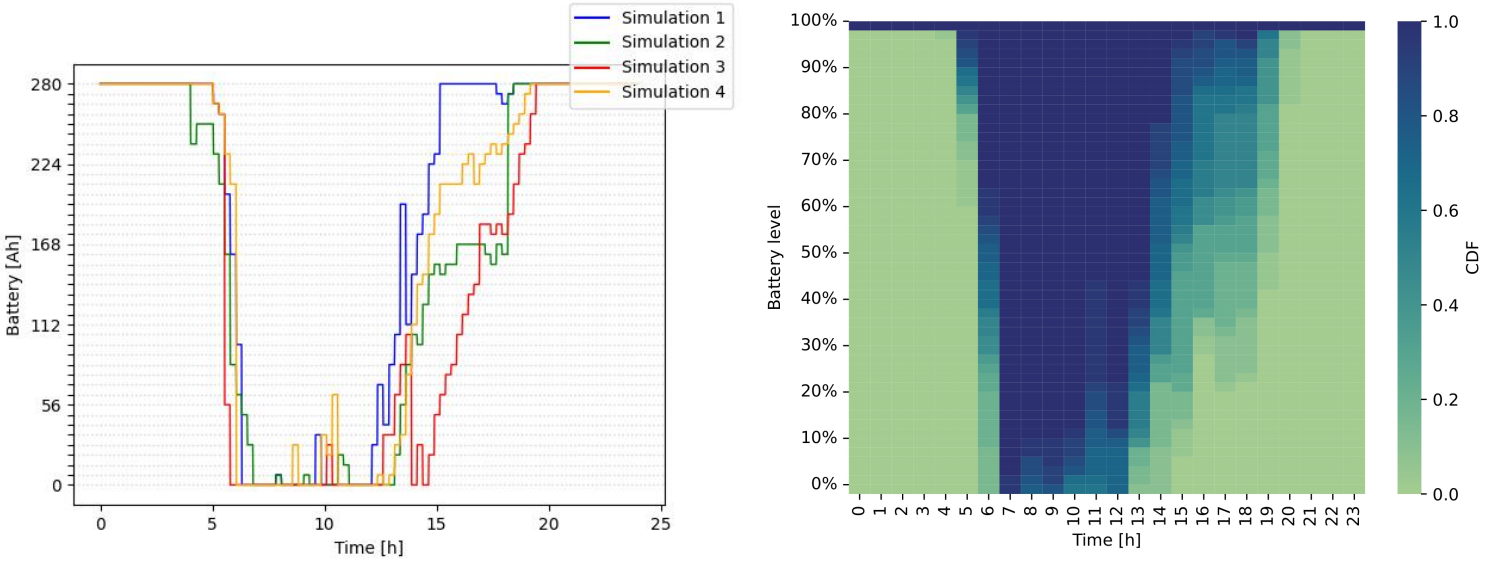
4.2.1 Model simulation and testing

This subsection focuses on discovering which one of the models proposed, DTMC or AR models, has a better performance when predicting the battery status. To this end, data from the PT is extracted. This data contains the battery status of a satellite located in 0

degrees of longitude rotating during 10 complete orbital periods. The battery status is computed taking into account the power received from the Sun and the power consumed when providing a certain rate to the users.

In the case of the DTMC model, the data is used to compute the transition probability matrices and these matrices are used to simulate the battery status of an entire orbital period. For the AR model, the ACF of this data is computed to predict the status of the batteries of the next full period.

Figure 4.4a illustrates 4 simulations of the predicted values done with a 41-state DTMC model. As mentioned in Section 3.2.1, the probability matrices and the respective simulations are computed with 5, 11, 21, 41, 51 and 101 number of states of the DTMC. These simulations can be found in Appendix B. Figure 4.4b shows the Cumulative Distribution Function (CDF) of the 41-state simulation. This CDF is computed individually for each Markov Chain of every hour using values obtained from a total of 10 predictions. For instance, at hour 0, the battery is almost certainly fully charged, while by hour 12, the battery is almost certainly below 40 %.



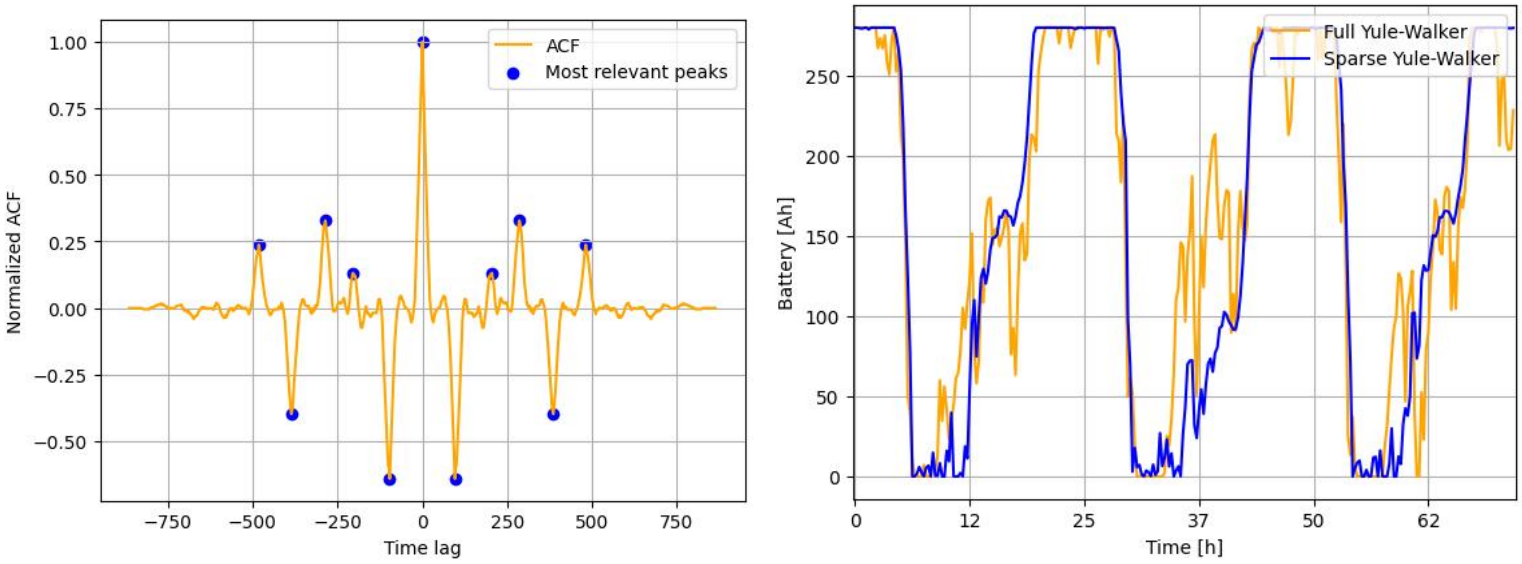
(a) 41-state simulations

(b) 41-state simulations CDF

Figure 4.4: 41-state DTMC model simulation

Regarding AR models, the ACF obtained from the simulated data is shown in Figure 4.5a. As a remark, the data used to compute this ACF has been differenced using as the period the Earth rotation and the resulting ACF has also been normalized. The orange line of the Figure corresponds to the complete ACF, used for Full Yule-Walker equations, and

the blue scattering on top are the most relevant peaks of the it, used for the Sparse Yule-Walker equations. The predicted data obtained with both approaches, Full and Sparse, is presented in Figure 4.5b. This graph corresponds to predicting 3 complete orbital periods. It is easy to appreciate that the results obtained by the Sparse Yule-Walker methodology are less affected by noise, as the similarity of the results between days is higher.



(a) ACF

(b) Predicted values

Figure 4.5: AR model

Comparison between models

In order to compare both approaches, DTMC and AR models, MSE is used. As mentioned in the beginning of the Section, the number of simulations from the Physical Twin used to extract the models is 10 days. After computing the models, predicted data for 10 complete orbital periods (10 days) is generated. For each day, the MSE of the difference between predicted and generated data is computed and a box plot combining the days is obtained.

Box plotting is introduced to have a better statistical understanding of the MSE values obtained for each realization. Usually, in the case of data prediction, the straightforward manner would be to choose the methodology that gives the lowest MSE on average. However, when the prediction is based on random factors, the performance of the model can have huge fluctuations. Figure 4.6 shows graphically a box plot.

- The orange line in the middle corresponds to the median of the data and the dotted green line, to the mean.

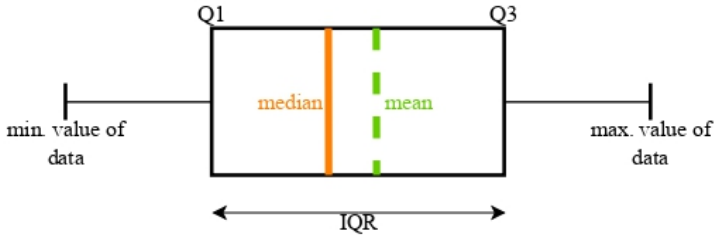


Figure 4.6: Boxplot

- The Quartile 1 (Q1) is the median of the lower half of the data and the Quartile (Q3) is the median of the upper half.
- The Interquartile Range (IQR) is the difference between Q1 and Q3.

When comparing box plots, there are two basic indicators to evaluate the results: the mean/median and the size of the box. As expected, the lowest mean/median indicates the lowest MSE. Nevertheless, the narrowest box plot is the most consistent one; the random factors generated by the models do not have much importance on the predictions and the MSE in the future realizations will not differ much from the one obtained.

Finally, Figure 4.7 shows the boxplots obtained for each one of the approaches.

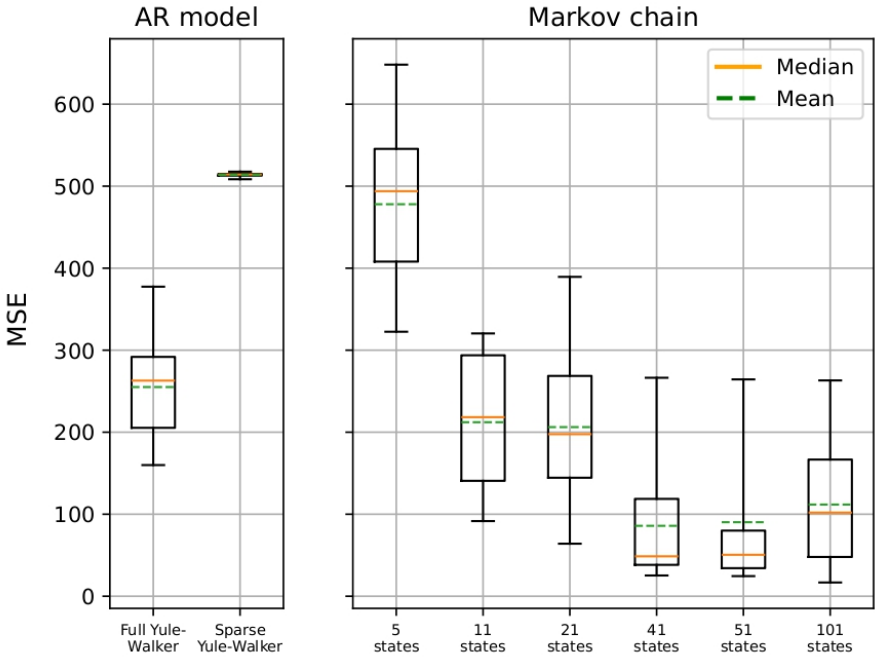


Figure 4.7: Predicted data MSE obtained using 10 simulations of real data

From that graph, it is possible to conclude that:

- The Sparse Yule-Walker model has a worse performance than the Full-Yule Walker one. This can be due to the fact of missing important correlations and oversimplifying the model, reducing model fidelity. However, the predicted error of the Sparse Yule-Walker model has less variability (lower variance) because the Full Yule-Walker may be overfitting. As a comparison between both AR models, in terms of robustness, it is better to choose Sparse Yule-Walker but, in terms of accuracy, the Full Yule-Walker behavior is better.
- Regarding the DTMC model, it is possible to see a decreasing tendency on the MSE as the number of states grows. According to the graph, the highest accuracy is obtained with the models of 41 and 51 states. Nevertheless, this outcome seems to be against the expected result; a major number of states should imply a decrease of the MSE and, in the graph, the MSE is higher with 101 states than with 51 states. This anomaly can be due to the fact of not having enough data for computing the transition matrices. The 101-state DTMC requires a larger amount of simulation data because all the states need to have representation. For that reason, MSE is recomputed in Figure 4.8 using 100 simulations of real data.

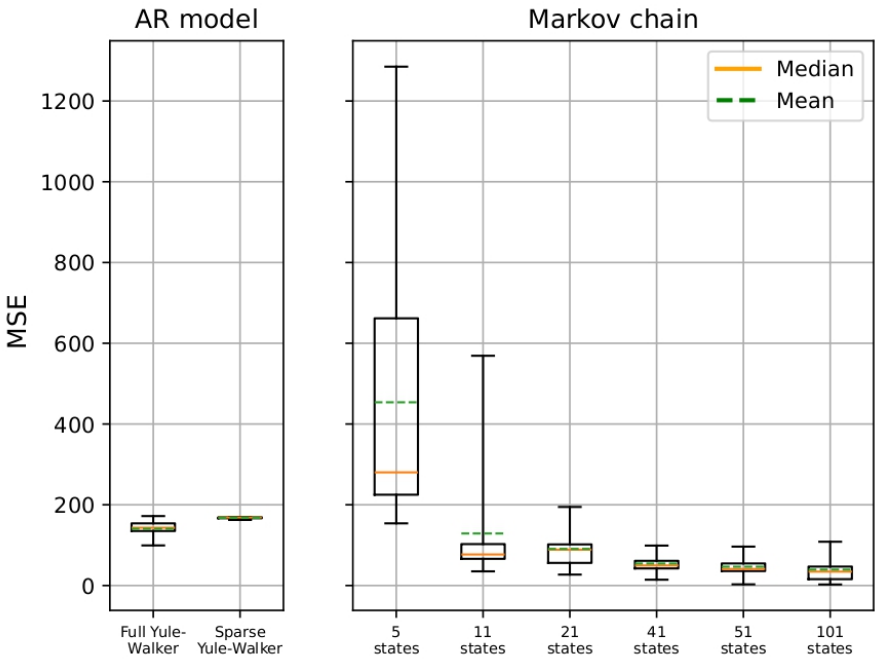


Figure 4.8: Predicted data MSE obtained using 100 simulations of real data

In this Figure, the decreasing trend of the MSE when using more states in the DTMC model is clearly noticeable. Apart from that, the MSE values obtained with the AR models

are also lower than the one obtained with 10 simulations. This behavior makes sense; the AR models work with autocorrelation values and with few samples, these estimates are noisy and biased, however, as the number of samples increases, the law of large numbers ensures that the sample autocorrelations converge to their true values. Regarding the behavior observed in the MSE obtained with 10 simulations, where the variance of the Full Yule-Walker was significantly large, here it is possible to see a more robust result, indicating a less overfitting model.

As a conclusion, the model chosen for the DT is the DTMC of 41 states. This decision has been motivated by two criteria:

1. DTMC models have obtained lower MSE values in comparison with AR models.
2. The lowest possible MSE value corresponds to the 101-state model. However, the difference between the last three models (41, 51 and 101 states) is barely noticeable and the 41-state matrices are easily manageable in terms of computing speed and memory administration.

4.2.2 Digital Twin service: Reinforcement Learning

The objective of the DT service is to ensure user connectivity while increasing the operational life of the batteries. In order to achieve it, this project proposes using RL to perform a time varying resource allocation. The goal is to assign satellites to certain coverage areas making sure they can handle the demanded rate without failing. A failure will be considered when the battery of a satellites drops below 0%, implying that the system is in outage. This section explains the different coverage areas, analyzes the problem feasibility and finally, shows the training procedure and the results obtained using RL.

Coverage areas

The results obtained in Section 4.1.1, using clustering, located the satellites in four deterministic positions, in the longitudes of -87, 14, 87 and 114. After placing the satellites there, it is possible to see that they have some overlapping coverage areas while other areas must be handled by individual satellites. Figure 4.9 shows the satellite positions and the total coverage area of the NTN. The Figure divides the total coverage area of the constellation in 8 areas. The first four areas correspond to mandatory zones that each satellite must cover, meaning Satellite 1 will always cover Area 1, Satellite 2 will cover Area 2, and so on. However, areas from 5 to 8, can be covered by more than one satellite and resource allocation needs to be used. Regarding the Figure, there is a small remark: the area resulting from the overlapping between area 1 and 4 is not taken into account because there is no human population living therefore it does not introduce traffic to the network. Table 4.1 summarizes the dependency between areas and satellites, showing the possible satellites for each area.

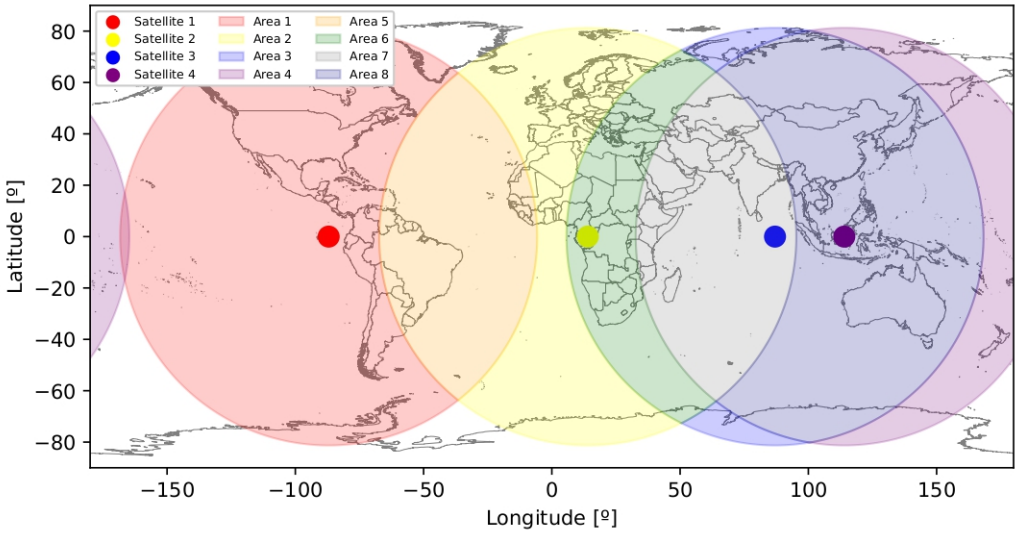


Figure 4.9: Satellite placement and coverage areas

Problem feasibility

As shown in the previous Table, not all satellites can cover every area of interest. For example, area 8 is only part of the footprint of satellites 3 and 4. In that case, it is essential to ensure that at least one of them can handle its mandatory area and the area of interest without discharging completely, meaning the satellite will have enough available energy and it will perform without failing.

In order to do so, the following experiment is conducted for each one of the shared or overlapping areas (from 5 to 8, both included):

1. The PT simulator is used to determine the maximum battery consumption in a simulation step, corresponding to 15 minutes, for each one of the satellites that can handle that overlapping area. Logically, the maximum battery consumption will happen when the rate required is the largest, corresponding to the worst-case load.

2. To the value obtained, it is necessary to add the battery consumed by the mandatory zone of the respective satellite. For example, if the possibility of covering area 5 by satellite 1 is being analyzed, the total battery consumption would be equivalent to the energy required to provide the maximum rate for areas 1 and 5 during 15 minutes.

3. Once the maximum battery consumption is known for each pair of area-satellite, the maximum battery recharging in a satellite is computed. In that case, the PT is used to calculate the maximum amount of energy that it can be stored during 15 minutes, based on solar exposure. The maximum amount corresponds to the satellite

Area	Satellite 1	Satellite 2	Satellite 3	Satellite 4
1	✓			
2		✓		
3			✓	
4				✓
5	✓	✓		
6		✓	✓	
7		✓	✓	✓
8			✓	✓

Table 4.1: Coverage assignment matrix showing which satellites are responsible for specific areas. A check mark (✓) indicates that the area is within the footprint of the satellite.

being in the orbital point closest to the Sun, when the solar panels are completely perpendicular to the Sun light and there is no eclipse effect.

4. For each overlapping area, at least one satellite must have a battery consumption value lower than the maximum recharging capacity.

Figure 4.10 shows the results obtained. The red line corresponds to the maximum battery charging during 15 minutes, around 60 Ah, and the gray doted line the relation between the rate required and the battery consumption associated to it during 15 minutes. Mathematically, this curve can expressed as in Equation 4.2 and it is extracted from the Shannon theorem and the relation between the power received and transmitted, Equations 4.1. The mathematical model is equivalent to the exponential fitting for the curve shown in the graph.

$$P_R(R) = P_N\left(1 - 2^{\frac{R}{B}}\right) \rightarrow P_T(R) = P_N\left(1 - 2^{\frac{R}{B}}\right) \cdot \alpha \tag{4.1}$$

$$\Delta C = \frac{P_T}{V} \cdot \frac{\Delta t}{3600}[Ah] \rightarrow \Delta C(R) = \frac{P_N\left(1 - 2^{\frac{R}{B}}\right) \cdot \alpha}{V} \cdot \frac{15 \cdot 60}{3600}[Ah] \tag{4.2}$$

In the image, each one of the colored dots corresponds to the rate consumed by a different satellite for a specific area. In that regard, areas 5, 6 and 7 can be handled by any of their available satellites within the battery capacity, consequently resource allocation can be applied. However, that is not the case of area 8, where both data points are above the limit and therefore, there is no satellite capable of handling it without exceeding the battery limitations.

With the goal of giving a feasible problem to the RL algorithm, Q-Learning, where all the satellites can operate without dropping all the battery and all the terrestrial zones are

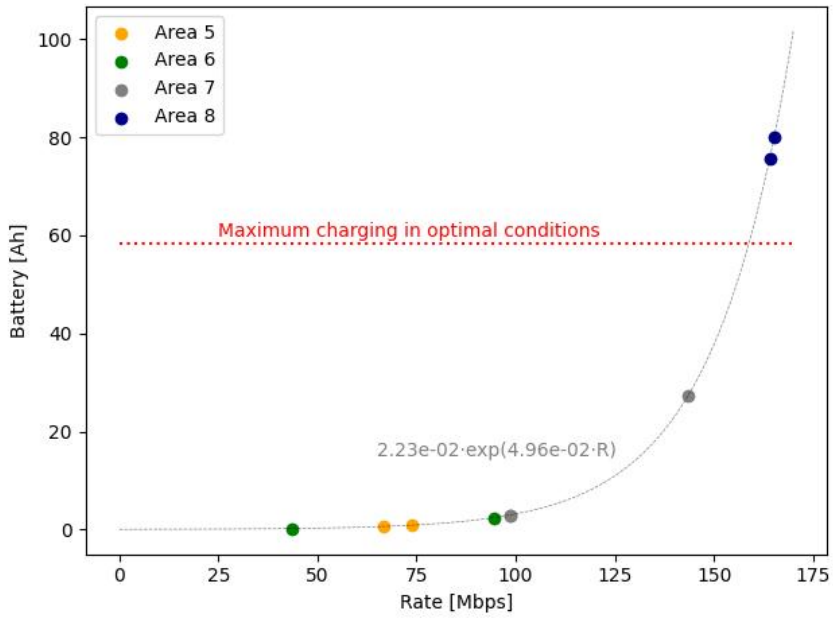


Figure 4.10: Relation between rate required and battery consumption

covered, a simultaneous coverage is proposed. In that case, the required rate by zone 8 will have the option of being satisfied by two satellites, 3 and 4, at the same time. This will avoid exceeding the battery capacities of one them because the requested power will be split.

Regarding the methodology for splitting the traffic, Round Robin algorithm is introduced [31, 17]. This methodology allows an fair distribution of workload between satellites. For simplicity, in this project, it is assumed that each one of the satellites will handle half of the rate requested in the area in case of splitting.

Reinforcement Learning environment and reward

The environment of the RL algorithm consists of a set of states and actions. Firstly, the states were defined in the previous chapter as the discretizations of the battery percentages. After choosing the 41-state DTMC as the model, the state space can be defined as the 41 possible states of each one of the batteries. Regarding the possible actions of the RL algorithm, they should include all the possible combinations of coverage, always ensuring that all the areas are covered. The total number of actions is 36 and they are summarized in Appendix C.

Finally, the reward proposed for the Q-Learning algorithm is as follows:

- Penalization of -100 if the battery drops below 0%. This part of the reward is oriented

to avoid having the system out of order.

- A high reward when the action taken allows having a small difference between battery status in order to have a low C-Rate. This is computed taking the difference between the current battery status and the predicted next one. In this case, two options are considered: a linear and a logarithmic approach. Both of them are shown in Figure 4.11, where having no difference is clearly awarded. The x-axis goes from 0 to 40 because that includes all the possible difference between states.

  The slope of the the linear approach, corresponding to the first design, has been decided after a trial-and-error process. If the slope were larger, the algorithm would not be able to learn because of not taking into account the penalization for battery deployment. In contrast, for smaller slopes, the algorithm ignored completely the C-Rate constraint. Using the maximum and minimum reward-values of the model (40 and 0, respectively), the logarithmic approach was designed to be in a similar interval.

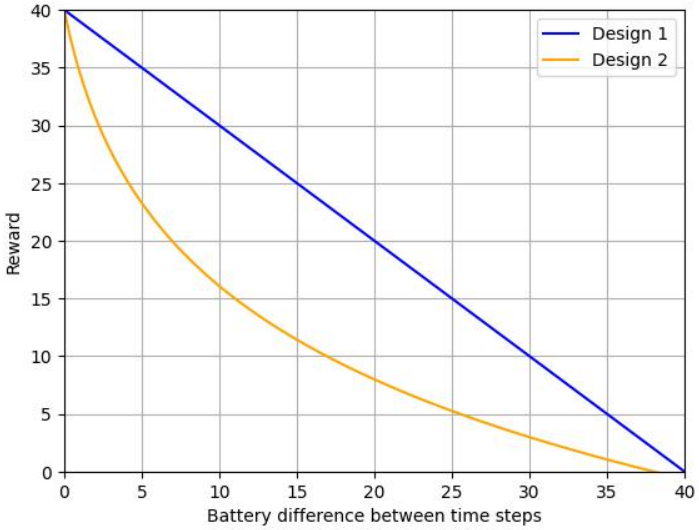


Figure 4.11: Reward regarding C-Rate

Training and Algorithm Performance

The RL algorithm, in particular Q-Learning, has been trained using predicted data from the Markov model. The learning parameters used are summarized in Table 4.2. As a remark, each one of the number of episodes corresponds to an entire rotation of the Earth.

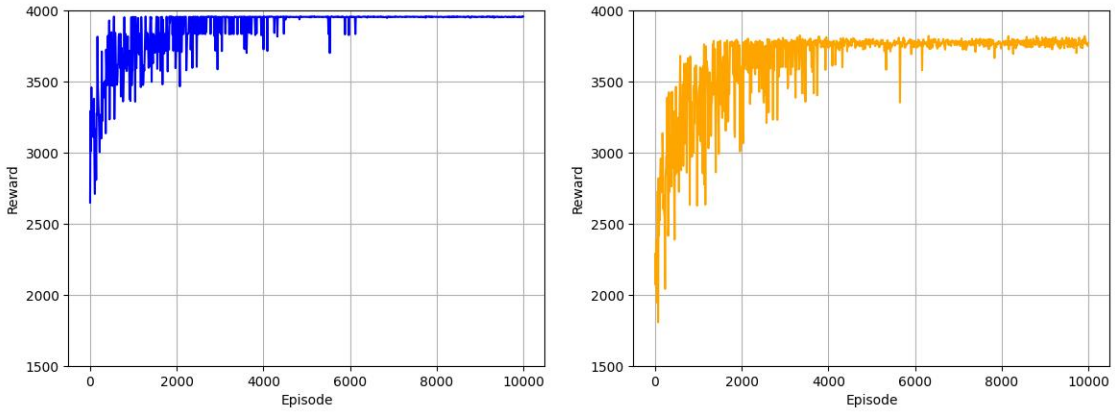
Parameter	Value
Number of episodes	10000
Learning rate	0.8
Discount factor	0.95
Exploration probability	0.5

Table 4.2: Learning parameters

The values chosen for this parameters are justified as follows:

- A large number of episodes is required to train a RL algorithm. It is found that with 10000, the model is able to learn. A larger number of episodes would just increase the training time.
- The values of the learning rate and the discount factor are the commonly used in research papers.
- It is found that a smaller exploration probability does not allow the model to learn and a larger one, does not focus enough in the exploitation. This value is found by trial-and-error. The exploration probability is also reduced for each episode.

Figure 4.12 shows the total reward per episode during training using the different designs. In both cases, the agent initially experiences large fluctuations but gradually learns to maximize cumulative reward, reaching near-optimal performance after approximately 7000 episodes. Comparing the designs, the first approach has a better performance regarding stability, the final reward values fluctuate less than the ones of the second approach. For that reason, that design is chosen as the final one.



(a) Reward obtained using design 1

(b) Reward obtained using design 2

Figure 4.12: Total reward per episode during training

The testing of the RL is also done with the Markov models due to code simplicity. The Q-Table obtained using the first reward design training is used to choose the action the constellation should take regarding resource allocation. It is tested for five complete orbital periods, corresponding to five days. The results in Figure 4.13 show how the algorithm is able to avoid dropping the batteries below 0% while keeping them stable and ensuring user connectivity. Each subplot represents one satellite, with battery capacity plotted over time in hours. Vertical red dashed lines indicate day boundaries (every 24 hours). The model is able to keep satellite 1 and 4 with full battery almost all the time and it introduces small fluctuations in satellites 2 and 3.

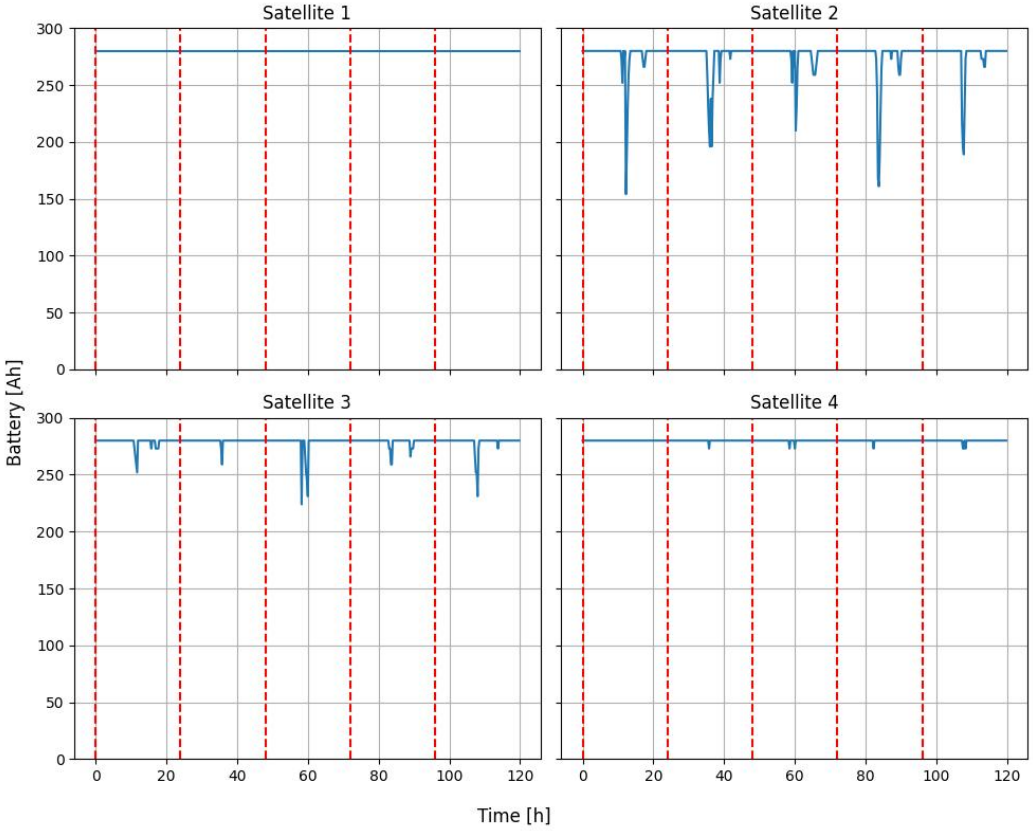


Figure 4.13: Battery levels of each satellite over a 5-day simulation

The results obtained with the RL algorithm are compared to a Constellation simulation of an entire Earth rotation where the resource allocation is performed randomly. The values obtained regarding the metrics in this case are as follows: the time for the system to get out of order corresponds to approximately 10 hours, when the first satellite drops the battery below 0% and the total lost throughput can be seen in Figure 4.14. As a conclusion, the DT service will be able to ensure connectivity to the users while taking into account

battery durability if the RL algorithm is used. It clearly fulfills the goals from the problem statements and it has an incredible performance regarding the metrics proposed.

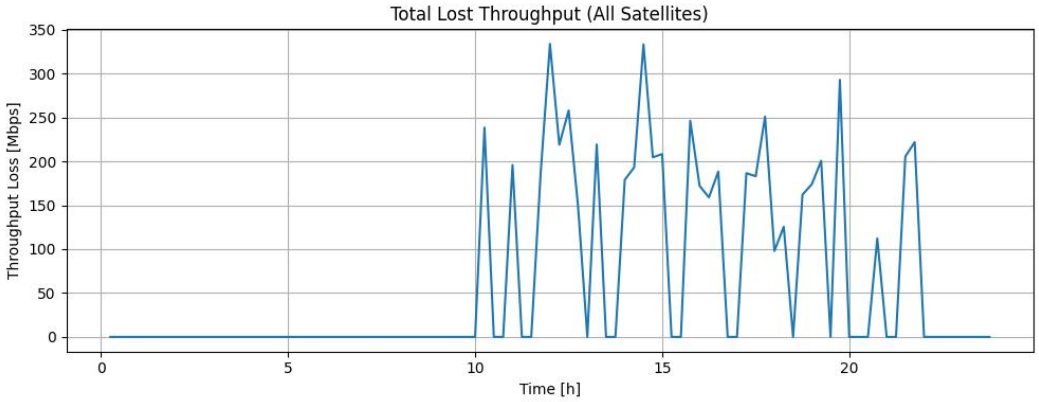


Figure 4.14: Total lost throughput from all satellites over a 24-hour period without using the Digital Twin service

Chapter 5

Conclusions

This project focused on developing a Physical Twin simulator of a Geostationary Orbit satellite constellation and the corresponding Digital Twin model and service, in order to answer the problem statement:*How can the battery status of satellites be accurately modeled with stochastic processes?/ How can battery models and AI algorithms be used within the Digital Twin scenario to extend the operational lifetime of the satellites?*

To begin with, a 4-GEO satellite simulator was developed in Python for generating realistic data on the battery status of the satellites. The battery charging was modeled following sunlight exposure and the discharging corresponds to the power demand from Earth users to maintain a specific data rate. For mimicking the human behavior, the rate requested was simulated as a stochastic process. This entire simulation framework is considered the Physical Twin of the project, the Cyber-Physical System that tries to be reproduced in the Digital Twin.

The data extracted from the Physical Twin simulator was used to calibrate and compare two different battery models for the Digital Twin; Discrete-Time Markov Chain and Auto-Regressive models. Firstly, five Markov models with different number of states were tested to verify the intuitive assumption that a larger number of states would imply higher accuracy. Secondly, a full and a sparse Yule-Walker Auto-regressive models were used to predict the battery status in the Digital Twin. The reasoning behind fitting the data into a sparse model lies in the attempt to reduce the effect of the noise introduced by the stochastic nature of power consumption. All models were compared on basis of accuracy between the data extracted from the Physical Twin and the predictions made by the Digital Twin, with the best results obtained using the Markov approach. Although the initial hypothesis regarding the benefit of a larger number of states was confirmed, the increase in accuracy became negligible beyond a 41-state model.

Once the Digital Twin had an accurate model of the battery's behavior, a service was im-

plemented to ensure long life expectancy in the Physical Twin. It was found that battery aging had a clear relationship with the Capacity-Rate, a measure that describes the rate at which a battery is charged or discharged, and, naturally, with avoiding states where the charge drops below 0%. In this context, a Reinforcement Learning algorithm was implemented to perform resource allocation, taking into account the current battery status and future predictions from the Markov model. The reward for the Reinforcement Learning model was based on minimizing fast battery aging. As a result, the model was able to keep the stability of the batteries. The effectiveness of the Digital Twin service was evaluated by comparing battery performance with and without its implementation, using two metrics: unmet throughput demand and time to depletion. In both cases, the use of the Digital Twin led to significant improvements; no throughput was lost, and battery depletion was effectively prevented over the observed time.

In conclusion, the project successfully developed a system comprising both a Physical and a Digital Twin for modeling the battery status of the satellites. Moreover, the goal of ensuring long battery life was achieved through the implementation of the Digital Twin service.

Bibliography

[1] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[2] Yannick Borthomieu. "Satellite lithium-ion batteries". In: *Lithium-ion batteries*. Elsevier, 2014, pp. 311–344.

[3] Masud Chaichian, Anca Tureanu, and Hugo Perez Rojas. *Basic concepts in physics*. Springer, 2021.

[4] Gidon Eshel. "The yule walker equations for the AR coefficients". In: *Internet resource* 2 (2003), pp. 68–73.

[5] European Space Agency. *Satellite frequency bands*. Accessed: 2025-05-19. 2013. URL: https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Satellite_frequency_bands.

[6] John Fitzgerald, Cláudio Gomes, and Peter Gorm Larsen. *The Engineering of Digital Twins*. 2024.

[7] Carole A Hill. "Satellite battery technology—A tutorial and overview". In: *IEEE Aerospace and Electronic Systems Magazine* 26.6 (2011), pp. 38–43.

[8] Kurt Hornik et al. "Spherical k-means clustering". In: *Journal of statistical software* 50 (2012), pp. 1–22.

[9] International Telecommunication Union (ITU). *Households with Internet Access at Home*. https://datahub.itu.int/data/?i=12047. Accessed: 2025-05-16. 2024.

[10] Muhammad Ali Jamshed et al. "A Tutorial on Non-Terrestrial Networks: Towards Global and Ubiquitous 6G Connectivity". In: *Foundations and Trends® in Networking* 14.3 (2025), pp. 160–253.

[11] Xu Jun, Wei Lu, and Gengxin Zhang. "Traffic modelling and simulation of broadband LEO satellite communication system". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 452. 4. IOP Publishing. 2018, p. 042082.

[12] Dong-Hyun Jung et al. "Modeling and analysis of GEO satellite networks". In: *IEEE Transactions on Wireless Communications* (2024).

[13] Steven Kay. *Intuitive probability and random processes using MATLAB®*. Springer Science & Business Media, 2006.

[14] Dihui Lai and Bingfeng Lu. "Autoregressive Model for Time Series as a Deterministic Dynamic System". In: *Predictive Analytics and Futurism* 7 (2017), pp. 7–9.

[15] Shengbo Eben Li. "Reinforcement learning for sequential decision and optimal control". In: (2023).

[16] Jiahao Liu et al. "DRL-ER: An intelligent energy-aware routing protocol with guaranteed delay bounds in satellite mega-constellations". In: *IEEE Transactions on Network Science and Engineering* 8.4 (2020), pp. 2872–2884.

[17] Paul D Mitchell, David Grace, and Tim C Tozer. "Analytical model of round-robin scheduling for a geostationary satellite system". In: *IEEE communications letters* 7.11 (2003), pp. 546–548.

[18] William J Nahm, Carter J Boyd, and Robert A Montgomery. "Satellite internet technology implementation for the practice of medicine and surgery". In: *The American Journal of Surgery* 225.5 (2023), pp. 941–942.

[19] Natural Earth. *1:10m Cultural Vectors*. `https://www.naturalearthdata.com/downloads/10m-cultural-vectors/`. Version 5.1.1. Accessed: 2025-05-16. 2022.

[20] Tho Manh Nguyen, A Min Tjoa, and Juan Trujillo. "Data warehousing and knowledge discovery: A chronological view of research challenges". In: *International Conference on Data Warehousing and Knowledge Discovery*. Springer. 2005, pp. 530–535.

[21] Hossein Pishro-Nik. *Introduction to probability, statistics, and random processes*. Kappa Research, LLC Blue Bell, PA, USA, 2014.

[22] Samad ME Sepasgozar. "Differentiating digital twin from digital shadow: Elucidating a paradigm shift to expedite a smart, sustainable built environment". In: *Buildings* 11.4 (2021), p. 151.

[23] *Solutions for NR to Support Non-Terrestrial Networks (NTN) (Release 16)*. Technical Report 3GPP TR 38.821 V16.0.0. [Accessed: 2025-05-16]. 3rd Generation Partnership Project (3GPP), Technical Specification Group Radio Access Network, 2019.

[24] Mauro Tropea and Floriano De Rango. "A comprehensive review of channel modeling for land mobile satellite communications". In: *Electronics* 11.5 (2022), p. 820.

[25] Jon Tucker et al. *NASA Technical Reports Server: NTRS*. Tech. rep. 20190001376. Accessed: 2025-05-16. NASA, 2019. URL: `https://ntrs.nasa.gov/citations/20190001376`.

[26] Jean Walrand and Pravin Pratap Varaiya. *High-performance communication networks*. Morgan Kaufmann, 2000.

[27] Shuai Wang et al. "Introduction to Satellite Communications". In: *Understanding Satellite Communications: The Stochastic Geometry Perspective*. Springer, 2024, pp. 1–18.

[28] Christopher JCH Watkins and Peter Dayan. “Q-learning”. In: *Machine learning* 8 (1992), pp. 279–292.

[29] Markus Werner and Gérard Maral. “Traffic flows and dynamic routing in LEO inter-satellite link networks”. In: *Proceedings of the 5th International Mobile Satellite Conference (IMSC)*. 1997, pp. 283–288.

[30] Marco A Wiering and Martijn Van Otterlo. “Reinforcement learning”. In: *Adaptation, learning, and optimization* 12.3 (2012), p. 729.

[31] Hénoïk Willot et al. “Satellite Communication Resources Management in a Earth Observation Federation of Constellations”. In: (2025).

[32] Yuan Yang et al. “Towards energy-efficient routing in satellite networks”. In: *IEEE Journal on Selected Areas in Communications* 34.12 (2016), pp. 3869–3886.

[33] Gökhan Yüksek and Alkan Alkaya. “Effect of the Depth of Discharge and C-Rate on Battery Degradation and Cycle Life”. In: *2023 14th International Conference on Electrical and Electronics Engineering (ELECO)*. IEEE. 2023, pp. 1–5.

Appendix A

Population maps

Figure A.1 shows the millions of inhabitants per country according to Natural Earth Data and Figure A.2 illustrates the percentages of households with Internet access at home. Finally, Figure A.3 corresponds to the map obtained after combining the previous one, showing the number of active users in millions for each cell grid.

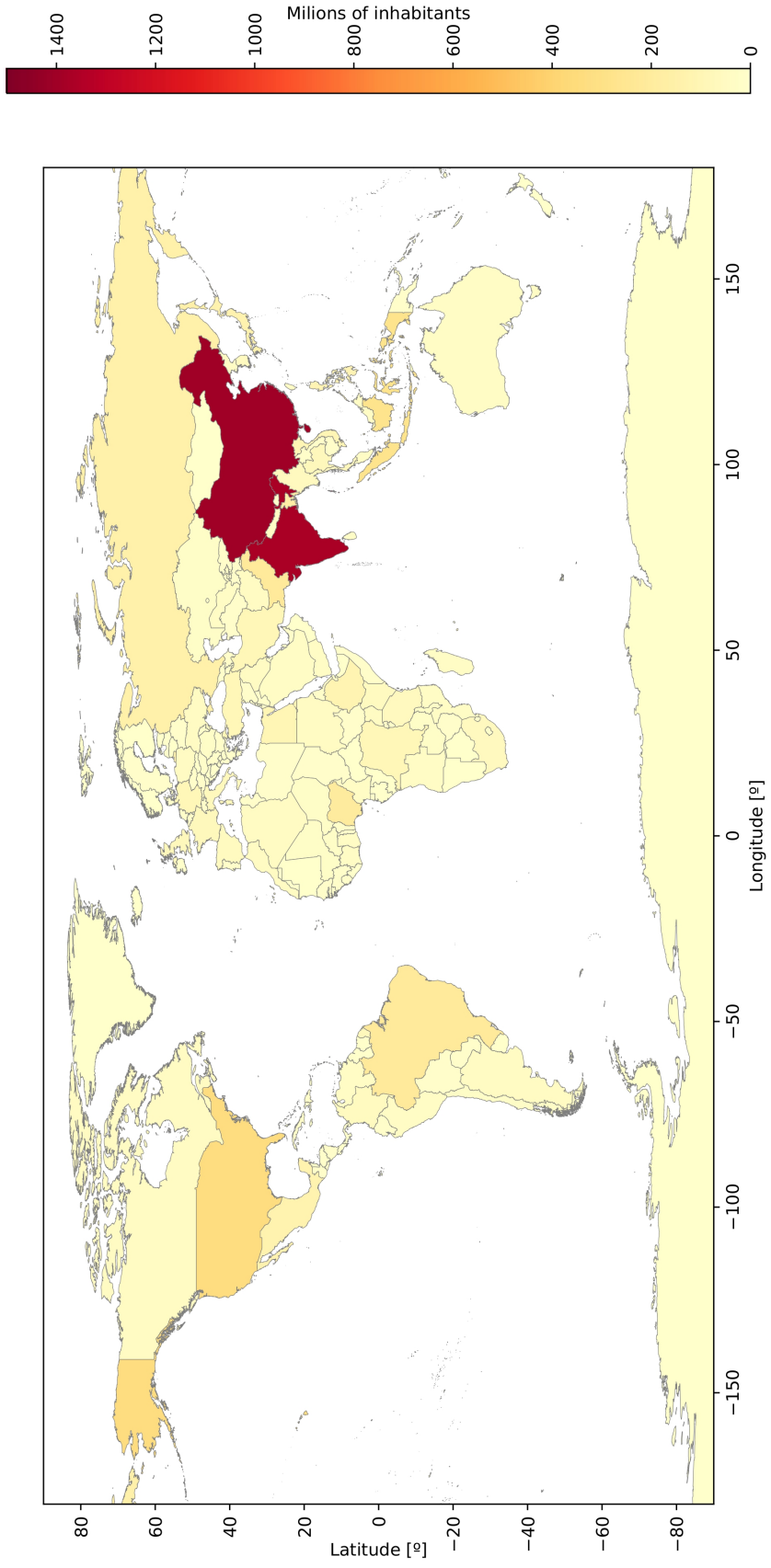


Figure A.1: Population map from Natural Earth Data

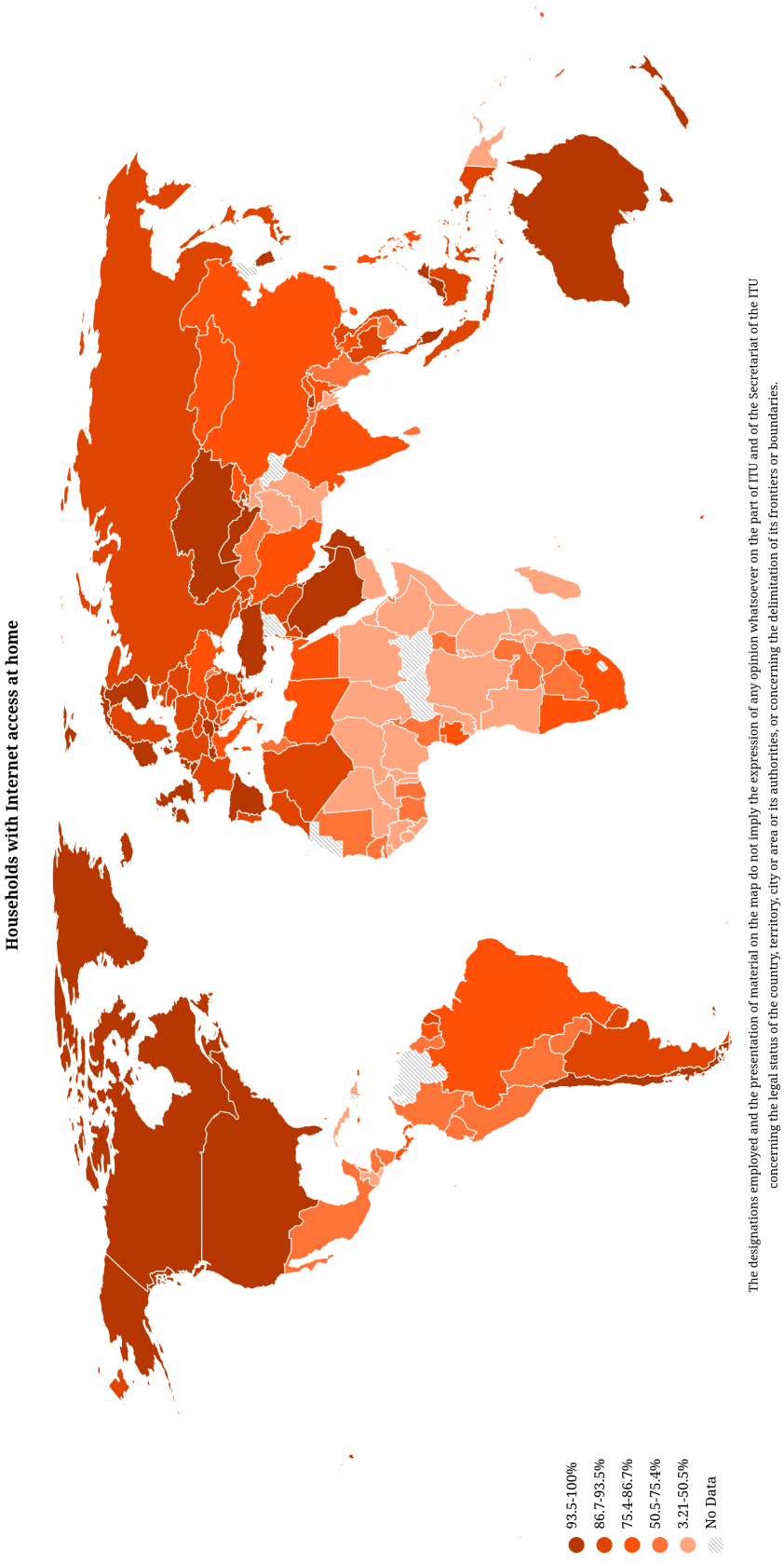


Figure A.2: Percentage of households with Internet access at home from ITU

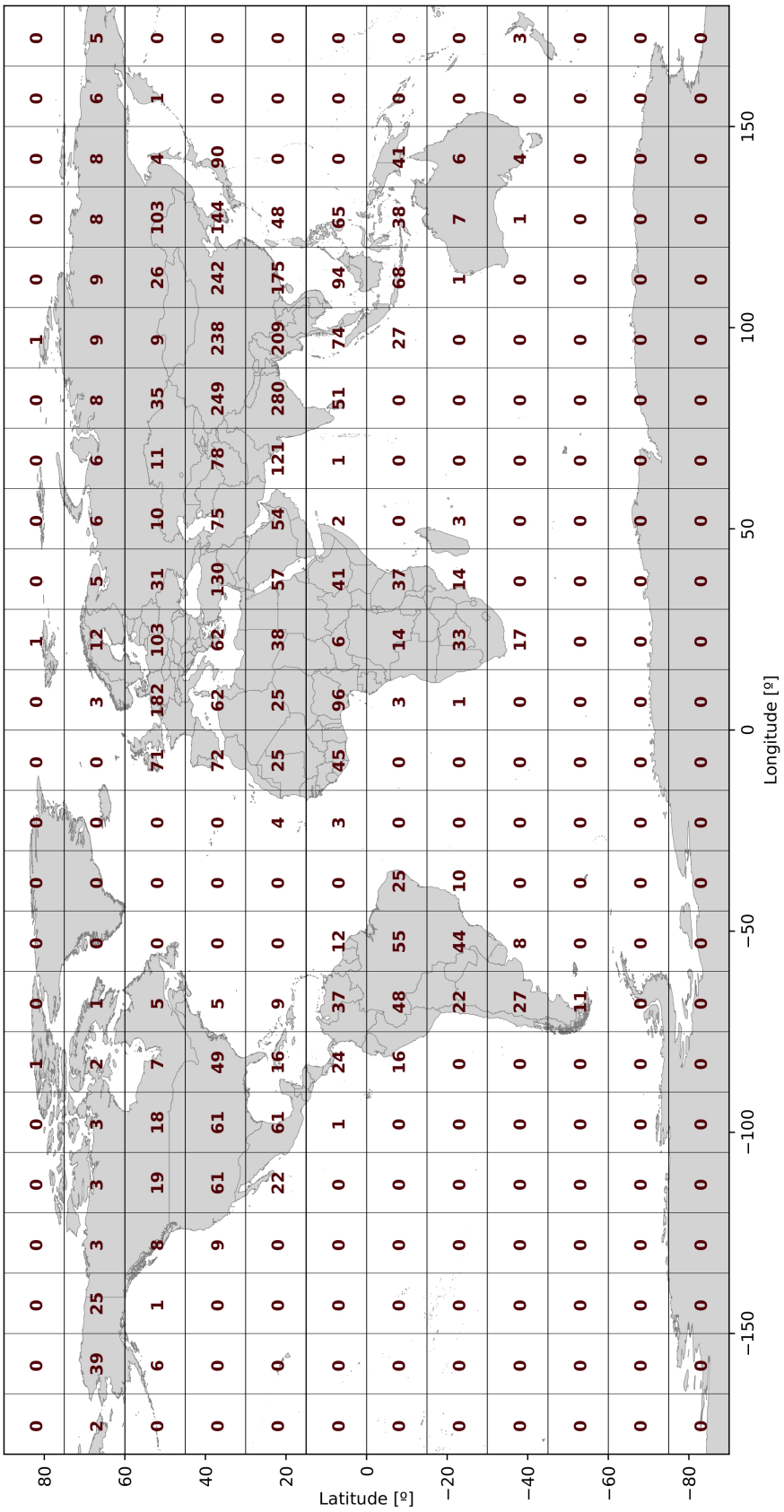
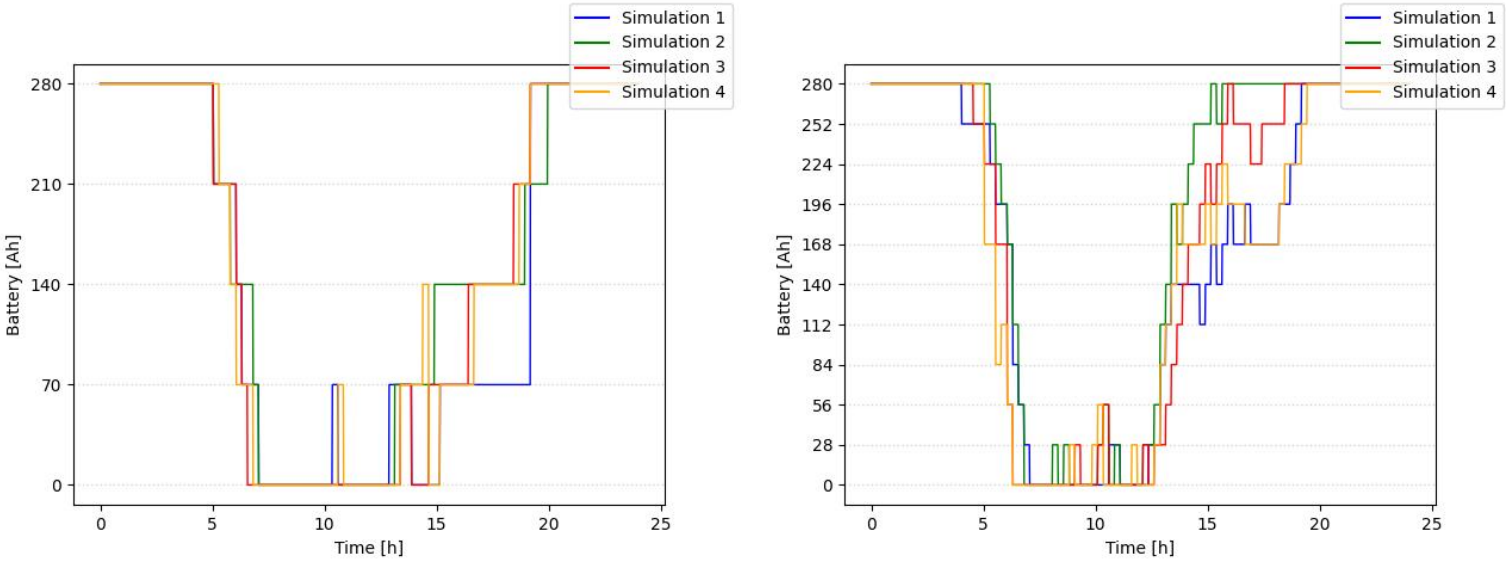


Figure A.3: Global map divided into a 12x24 grid showing the number of users (in millions) per cell

Appendix B

Complete simulations of the Discrete-Time Markov Chain models

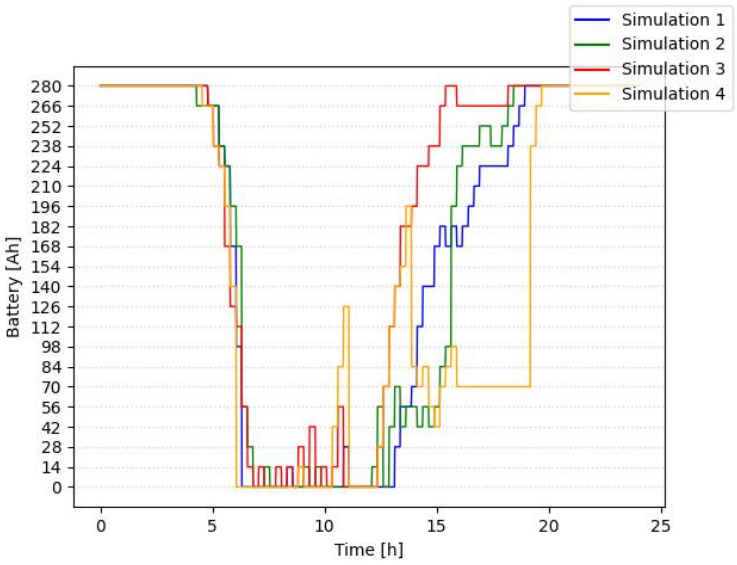
Figure B.1 shows 4 simulations of the battery of a satellite located in the Greenwich meridian for the different number of states proposed (51, 11, 21, 41, 51, and 101) for the DTMC model.



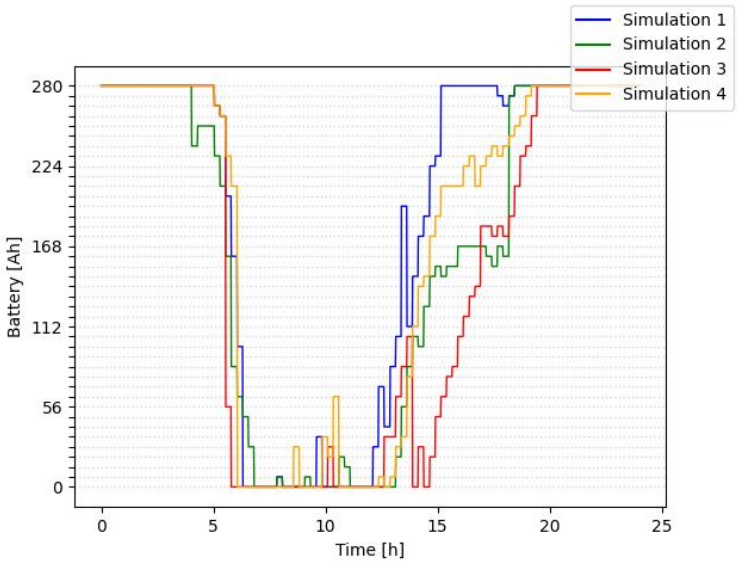
(a) 5-state simulations

(b) 11-state simulations

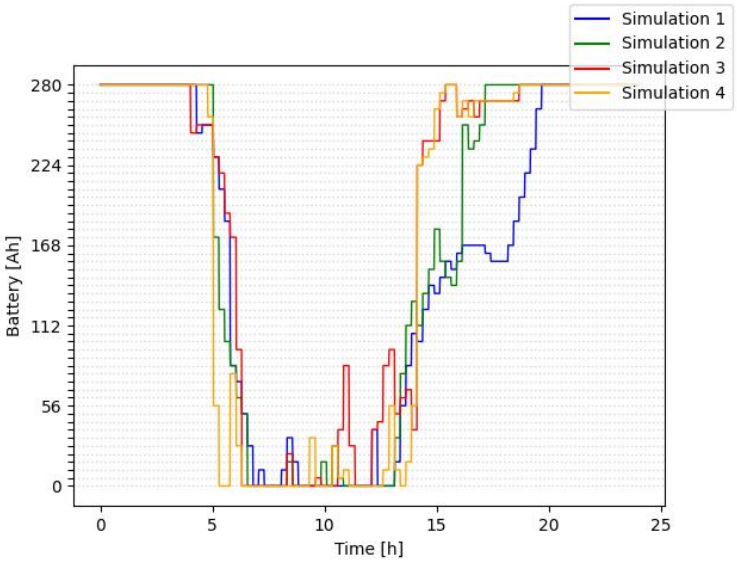
Figure B.1: DTMC – simulations overview (1/2)



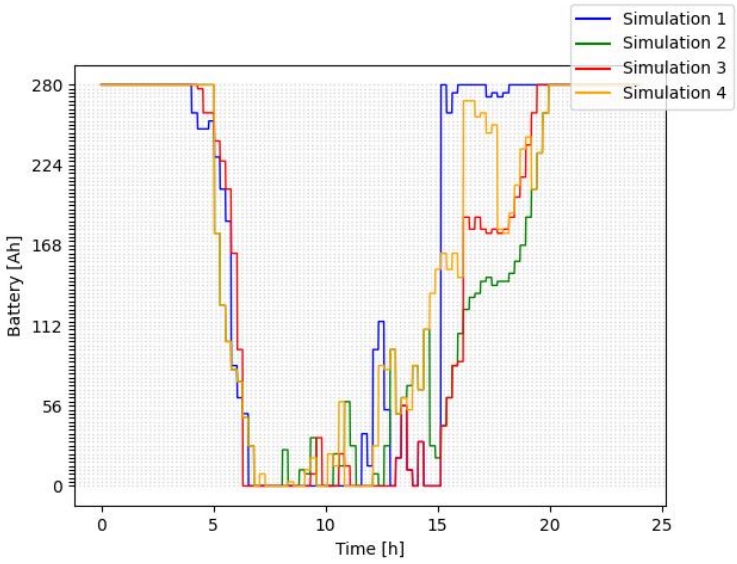
(c) 21-state simulations



(d) 41-state simulations



(e) 51-state simulation



(f) 101-state simulation

Figure B.1: DTMC – simulations overview (2/2)

Appendix C

Possible actions of the Reinforcement Learning algorithm

Table C.1 includes all the possible combinations within overlapping areas; the mandatory areas of each one of the satellites are not included. It is important to notice that in the case where area 8 is part of satellite 3 and 4, the user demand will be shared among them.

Action	**Coverage per Satellite**			
	Satellite 1	**Satellite 2**	**Satellite 3**	**Satellite 4**
0	5	6, 7	8	
1	5	6, 7	8	8
2	5	6, 7	8	8
3	5	6	7, 8	
4	5	6	7, 8	8
5	5	6	7	8
6	5	6	8	7
7	5	6	8	7
8	5	6	1	7, 8
9	5	7	6, 8	
10	5	7	6, 8	8
11	5	7	6	8
12	5		6, 7, 8	
13	5		6, 7, 8	8
14	5		6, 7	8
15	5		6, 8	7
16	5		6, 8	7, 8
17	5		6	7, 8
18		5, 6, 7	8	8

19		5, 6, 7	8	
20		5, 6, 7	1	8
21		5, 6	7, 8	8
22		5, 6	7	8
23		5, 6	7	8
24		5	6	7, 8
25		5, 6	8	7, 8
26		5, 6	1	7, 8
27		5, 7	6, 8	
28		5, 7	6, 8	8
29		5, 7	6	8
30		5	6, 7, 8	
31		5	6, 7, 8	8
32		5	6, 7	8
33		5	6, 8	7
34		5	6, 8	7, 8
35		5	6	7, 8

Table C.1: Possible actions of the RL algorithm