University of Cadi Ayyad

Faculty of sciences and technology - Marrakech
Department of Applied Physics

Engineering of Renewable Energies and Electric Mobility
Electrical Engineering

Internship report

Autonomous UGV for Environmental Monitoring with Solar Energy and AI Optimization

Prepared by:
Mourad Najjay
Hicham Elidrissi
Hafsa Korou
Hajjar Izelgue
Amina Bassou
Zakarya Ouiazzane
Mohamed Elghazi

Supervised by:
Mr. Anas SKITI

Academic year 2024–2025

Dedication

We dedicate this project to our families for their unwavering support and encouragement.

Our heartfelt thanks go to our mentors and colleagues for their invaluable guidance and insights.

Your support has been crucial in bringing this project to fruition.

Acknowledgement

First, we express our heartiest thanks and gratefulness to almighty Allah for His divine blessing in making us able to complete this project successfully.

We would like to thank the following people for their help in the production of this project. We are deeply indebted to our supervisor **Mr. ANAS SKITI**, [CEO of UXV Center]. We are profoundly grateful to him, for the opportunity to work under his guidance. His unwavering support and trust have been crucial throughout this endeavor. Without his help and support throughout this project, it would not have been possible. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, and reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

Our special thanks go to the manager, **SOUGRATI Ismail**, for his invaluable assistance with various aspects of the project.

We are deeply thankful to all our faculty members for their invaluable guidance and support. We also extend our sincere gratitude to everyone who contributed to the successful completion of our project.

Finally, we would like to acknowledge with heartfelt appreciation the unwavering support and patience of our parents.

Abstract

Our project is divided into two main parts. The first part focuses on researching and implementing techniques to optimize the solar energy used to power an autonomous ground vehicle (UGV). One solution involves training a machine learning model capable of predicting the best orientation angles for the solar panel to maximize the amount of captured solar radiation. Another solution relies on the use of the Python library pvlib, which allows direct obtaining of the orientation angles based on the vehicle's position data and the time of day.

The second part of the project focuses on developing an interactive interface to monitor data related to solar energy production, vehicle performance, as well as environmental data from the onboard sensors. Additionally, this interface will offer the possibility to generate daily reports and user-defined alerts.

Résumé

Notre projet se divise en deux grandes parties. La première partie ayant pour objectif la recherche et l'exploitation de techniques permettant l'optimisation de l'énergie solaire utilisée pour alimenter un véhicule terrestre autonome. Une première solution consiste à entraîner un modèle d'apprentissage automatique capable de prédire les meilleurs angles d'orientation du panneau solaire afin de maximiser la quantité de rayonnement solaire capté. Une deuxième solution repose sur l'utilisation de la bibliothèque pvlib de Python, qui permet de calculer directement les angles d'orientation en fonction des données de position du véhicule et de la période de la journée.

La deuxième partie du projet se concentre sur le développement d'une interface interactive permettant de suivre les données relatives à la production d'énergie solaire, aux performances du véhicule, ainsi qu'aux données environnementales provenant des capteurs embarqués. De plus, cette interface offrira la possibilité de générer des rapports quotidiens et des alertes personnalisées par l'utilisateur.

Contents

Contents i

List of Figures iv

1 Generalities 3

1.1 Introduction 4
1.2 Company presentation 4
1.2.1 Creation and Launch of UXV Center 4
1.2.2 Geographical Location and Infrastructure . . . 5
1.2.3 Services Offered by UXV Center 6
1.2.4 Organization 7
1.3 Project overview 7
1.3.1 Project Context 7
1.3.2 Project Description and Specifications 8
1.4 Essential tools and skills 9
1.4.1 Scrum method 9
1.4.2 Software tools 11
1.5 Task organisation and distribution 12
1.5.1 GANTT Chart 12
1.6 Conclusion 14

2 Overview of Solar Energy 15

2.1 Introduction 16
2.2 Solar Energy and Its Various Applications 16
2.3 From Cell to Photovoltaic Panel 18
2.4 The Different Materials Used in Photovoltaic Panel Manufacturing 19
2.5 The Principle of Photovoltaic Conversion 21
2.5.1 History of photovoltaic effect 21

2.5.2 Photovoltaic effect 23
2.6 Factors Influencing Solar Panel Efficiency (orientation) 25
2.7 Solar Trackers: An Optimal Solution to Maximize Solar Panel Performance 26
2.7.1 Introduction 26
2.7.2 Solar Tracking and PV Panel Efficiency 27
2.7.3 Design of the Proposed Solar Tracking System 30
2.8 Conclusion 30

3 Machine Learning Essentials **31**
3.1 Introduction 32
3.2 Foundations of Machine Learning 32
3.2.1 Definition of Machine Learning 32
3.2.2 Importance of Machine Learning 32
3.3 The Different Types of Machine Learning 33
3.3.1 Supervised Learning 33
3.3.2 Unsupervised Learning 34
3.3.3 Reinforcement Learning 34
3.4 Types of Machine Learning Problems 34
3.4.1 Regression problems 34
3.4.2 Classification problems 34
3.4.3 Clustering problems 35
3.5 Fundemental Steps for Solving a Machine Learning Problem 35
3.5.1 Define the Problem 35
3.5.2 Collect and Prepare Data 35
3.5.3 explore and analyze data 36
3.5.4 Selecting appropriate model based on the problem 36
3.5.5 Split the Data 36
3.5.6 Train the model 37
3.5.7 Evaluate the Model: 37
3.5.8 Conclusion 37

4 AI Solutions for Optimizing Panel Orientation **39**
4.1 Introduction 40
4.2 Optimizing panel orientation using a machine learning model 40

4.2.1 Dataset preparation 40
4.2.2 Model selection 43
4.2.3 Model training and evaluation 50
4.3 Optimizing panel orientation using pvlib library . . . 51
4.3.1 Introduction 51
4.3.2 What is pvlib ? 51
4.3.3 Pvlib usage in optimizing panel orientation . . 53
4.4 Interpretation and results exploitation 55
4.4.1 Light Sensor Integration for Efficient Panel Orientation 55
4.4.2 Adaptation of the solar panel orientation to the vehicle's movement 56
4.5 Conclusion 58

5 Real-Time Monitoring with Interactive Dashboards 59
5.1 Introduction 60
5.2 Designing the User Interface with Streamlit 60
5.2.1 Objectives of the Interface 60
5.2.2 Code File 61
5.2.3 Developing the API Infrastructure 62
5.2.4 Solar Panel Monitoring 64
5.2.5 UGV Monitoring 65
5.2.6 Environmental Data 66
5.2.7 Reports and Alerts 67
5.2.8 Overview of the Dashboard 67
5.3 Conclusion 68

List of Figures

1.2.1 Headquarters of the host company UXV Center. . . . 5
1.2.2 Company Hierarchy. 7
1.5.3 Sprint Management in Jira. 12
1.5.4 Task Organization for the First Four Weeks. 13
1.5.5 Task Organization for the Final Four Weeks. 13

2.3.1 From a solar cell. 19
2.5.2 Monocrystalline structure. 23
2.5.3 A diagram showing the photovoltaic effect. 24
2.7.4 Principle of the single-axis solar tracking system . . . 28
2.7.5 Solar tracker. 29

4.2.1 Azimuth and Tilt angles. 41
4.2.2 dataset before preprocessing. 42
4.2.3 Cleaned Dataset Post-Preprocessing. 43
4.2.4 Scikit-learn estimators guide. 44
4.2.5 Optimizing performance using Ensemble Learning. . . 45
4.2.6 comparison between Bagging and Boosting algorithms. 47
4.2.7 The architecture of stacking algorithms. 47
4.2.8 Fundamental elements of a decision tree. 49
4.3.9 pvlib logo. 52
4.3.10 simulation of the UGV's geographic coordinates. . . . 54
4.3.11 Using the get_solarposition function to determine the panel orientation. 55
4.4.12 function for improving energy efficiency using light sensors. 56
4.4.13 image of classic gyroscope. 57
4.4.14 MPU6050 gyroscope and accelerometer. 57
4.4.15 simulation code. 58
4.4.16 simulation code. 58

5.2.1 Dashboard showing the different sections. 61
5.2.2 Dashboard showing the different sections. 64
5.2.3 Real-time graphs and metrics related to solar energy production. 65
5.2.4 UGV Monitoring part. 66
5.2.5 Real-time graphs and metrics related to solar energy production. 66
5.2.6 Battery alert. 67

General introduction

In the rapidly advancing field of technology, vehicles have consistently led the way in innovation, evolving from early steam engines to today's sophisticated electric and autonomous systems. Unmanned Ground Vehicles (UGVs) represent a significant leap forward, transitioning from military and hazardous environments to a diverse range of applications, including environmental monitoring. Among the most promising advancements in UGV technology is the integration of solar energy. This shift from traditional battery and fuel sources to solar power not only extends operational time but also aligns with global sustainability goals by reducing reliance on non-renewable resources.

In the context of renewable energy, optimizing solar energy harvesting for autonomous systems poses both a challenge and an opportunity. This project focuses on enhancing the solar energy efficiency of UGVs by integrating state-of-the-art solar panels, complemented by artificial intelligence (AI) to manage solar panel orientation . AI algorithms will be employed to analyze sunlight exposure, dynamically adjust panel orientation, and predict optimal charging times based on real-time weather conditions.

This report is structured to provide a comprehensive overview of the project's objectives and methodologies. Chapter 1 introduces the foundational concepts relevant to UGVs and solar energy. Chapter 2 explores solar energy principles and their application to UGVs. Chapter 3 covers essential machine learning concepts necessary for implementing AI solutions. Chapter 4 focuses on specific AI strategies for optimizing solar panel orientation, while Chapter 5 emphasizes the importance of real-time monitoring and interactive dashboards for proactive management of the solar energy system. By enhancing solar energy harvesting capabilities, this project aims to improve UGV performance and sustainability, offering a new approach to environmental monitoring and data collection.

Chapter 1

Generalities

1.1 Introduction

In a Morocco experiencing rapid development focused on innovation and digital transformation, an ecosystem conducive to the emergence of innovative ideas is taking shape.Within this dynamic environment, UXV Center stands out as a major player, providing an ideal framework for the development and realization of engineering and innovative system projects. This chapter will focus on a detailed presentation of UXV Center as well as the general context surrounding our project. To conclude this chapter, we will outline the objectives of our internship, describing the methods and tools used to achieve them.

1.2 Company presentation

1.2.1 Creation and Launch of UXV Center

UXV CENTER GmbH, established in Marrakech since its inception in 2018, stands out as a global leader in the production of UXVs and software solutions 1.2.1.

The company is involved in every stage of the UXV technology process, from designing hardware and software components to pilot training, as well as testing and certification. This commitment reflects its dedication to innovation and quality. The main objective of UXV CENTER is to develop tailored solutions that meet the specific needs of its clients, leveraging strong technical expertise and high creativity. With a diverse team of UXV specialists, the company provides a comprehensive response to the technical requirements of its clients and partners.

Figure 1.2.1: Headquarters of the host company UXV Center.

1.2.2 Geographical Location and Infrastructure

The strategic location of UXV Center GmbH at Ottigenbühlring 5 in Ebikon, Switzerland, is crucial for the company. As the hub of its operations, this site provides an environment conducive to the innovation and collaboration essential in the UXV technology field. Switzerland is renowned for its excellence in research and development, offering UXV Center GmbH privileged access to a pool of skilled talent and cutting-edge technological infrastructure. Additionally, the country's political and economic stability bolsters the confidence of business partners and investors. Furthermore, the presence of a facility in Marrakech, Morocco, signifies a strategic expansion into new markets and greater geographical diversification. This location offers unique advantages, such as increased access to emerging markets, a skilled workforce, and competitive operational costs. Moreover, its proximity to North Africa enhances business relations with this dynamic region. By combining these two locations, UXV Center GmbH strengthens its global presence and reinforces its position as a major player in the technology sector. The synergy between these sites provides growth and development opportunities that enable the company to stay at the forefront of innovation and meet the evolving needs of the global UXV technology market.

1.2.3 Services Offered by UXV Center

Unmanned vehicles, or UXVs (Unmanned X-Vehicles), are highly versatile machines capable of providing valuable information to professionals across various fields. Their utility lies particularly in their ability to collect data in areas deemed too dangerous or difficult to access for humans. With a combination of increased flexibility and reduced capital and operational costs, unmanned vehicles have the potential to transform sectors as diverse as urban infrastructure management, agriculture, and oil and gas exploration, among others. UXV Center GmbH offers manufacturing and services involving three types of UXVs:

A. UAVs (Unmanned Aerial Vehicles)

UAVs are unmanned aerial vehicles that offer exceptional flexibility for data collection and mission execution in environments that are difficult to access or dangerous for humans. Their ability to cover extensive areas and capture high-resolution images makes them essential tools in fields such as mapping, environmental monitoring, search and rescue, as well as security and surveillance.

B. UGVs (Unmanned Ground Vehicles)

UGVs are autonomous ground robotics systems that operate without the presence of a human operator on board. They are used in a variety of applications, particularly in dangerous or hard-to-reach environments. Their capabilities are utilized for performing complex, monotonous, or high-risk tasks. Applications include the detection and neutralization of improvised explosive devices (IEDs), border surveillance, mining exploration, and many more.

C. USVs (Unmanned Surface Vehicles)

USVs are unmanned surface vehicles controlled remotely by an operator on land or aboard another vessel. They play a crucial role in various applications, including maritime security, waterway monitoring, oceanographic data collection, and military reconnaissance. Their use helps to minimize risks to personnel by maintaining a safe distance from potentially dangerous or hostile situations.

D. Batteries and Computer Boards

UXV Center GmbH offers high-quality batteries featuring numerous advantages and based on LiFePO4 technology, which stands for Lithium Iron Phosphate. Additionally, the company provides intelligent computer boards capable of storing and processing data in real time with high precision, while reducing both system size and computation time.

1.2.4 Organization

The organizational chart below represents the structure and hierarchy of the host company. See figure 1.2.2.

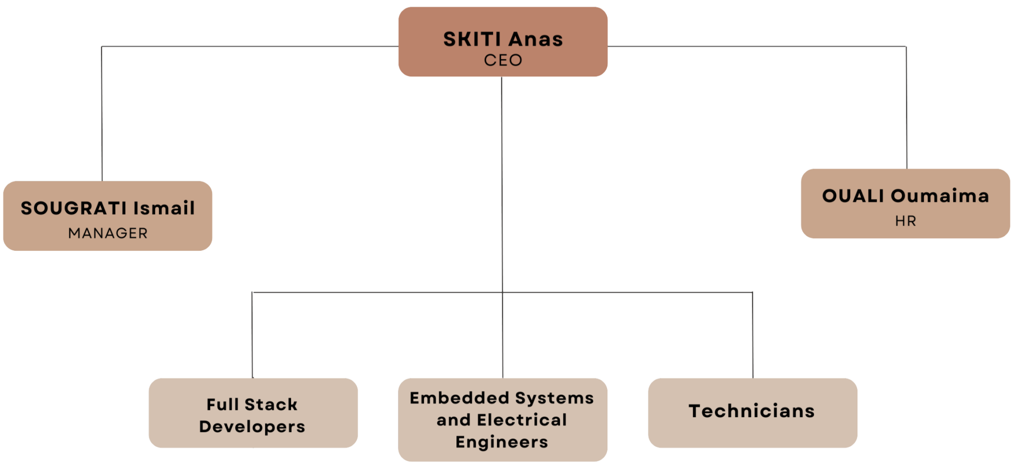


Figure 1.2.2: Company Hierarchy.

1.3 Project overview

1.3.1 Project Context

As concerns about environmental protection and sustainable resource management rise, UXV Center, a leader in autonomous vehicle technology, is embarking on a pioneering project to develop an autonomous unmanned ground vehicle (UGV) for environmental monitoring. This initiative aligns with UXV Center's commitment to innovation, integrating advanced artificial intelligence (AI) and solar power technologies to enhance vehicle performance and sustainability. The project aims to deploy UGVs capable of continuous real-time surveillance in diverse and complex environments ensuring precise monitoring, minimizing environmental impact through solar energy, and leveraging AI to optimize routes and adapt to changing conditions. This approach not only addresses contemporary ecological challenges but also positions UXV Center at the forefront of cutting-edge, eco-friendly solutions.

1.3.2 Project Description and Specifications

The project focuses on developing and implementing an advanced system to optimize solar energy harvesting for an autonomous Unmanned Ground Vehicle (UGV) equipped with solar panels. The primary aim is to enhance the efficiency of solar energy collection, thereby extending the operational duration of the UGV. This involves integrating a sophisticated energy management system with AI capabilities to maximize the benefits of solar power.

Objectives

1. Energy Management :
   - Solar Panel System: Integrate a high-performance solar panel system onto the UGV to capture solar energy effectively.
   - AI Optimization: Utilize AI to optimize the angle and orientation of the solar panels to ensure maximum exposure to sunlight and manage the battery charging process efficiently.
2. Data Analysis and Optimization :
   - Dynamic Adjustment: Implement AI algorithms to analyze real-time sunlight exposure data and dynamically adjust the orientation of the solar panels for optimal energy collection.
   - Predictive Analytics: Develop a predictive system that forecasts the best times and locations for charging based

on weather conditions and sunlight availability, improving energy utilization and storage.

Project specifications

The system is composed of various interconnected components designed to simulate, process, and utilize data for optimizing solar energy harvesting. The key components include:

- Data Source and Simulation : Produce simulated sensor data that accurately reflects the behavior and measurements of a real UGV.
- Data Processing and Storage : Goal: Process the simulated data by cleaning, formatting, and organizing it into a structured format for effective storage.
- Artificial Intelligence (AI) and Modeling : Develop and train AI models to determine the optimal orientation of solar panels by analyzing historical data.
- System Control and Decision Making : Utilize AI model predictions to dynamically adjust the orientation of solar panels.
- Visualization and Monitoring : Create visualizations to monitor and analyze system performance and the orientation of solar panels.

1.4 Essential tools and skills

To effectively embark on and advance this project, it is crucial to possess essential tools and skills, such as proficiency in the Scrum methodology for streamlined team collaboration, along with a solid foundation of relevant skills and knowledge necessary for project development and implementation.

1.4.1 Scrum method

Scrum is a management framework that teams use to self-organize and work towards a common goal. It outlines a set of meetings,

tools, and roles to facilitate effective project execution. Similar to a sports team training before a big match, Scrum practices enable teams to self-manage, learn from experience, and adapt to change. Development teams use Scrum to tackle complex problems cost-effectively and sustainably.
Certain principles and values define the Scrum methodology:

1. Principles of Scrum for Project Success

    - Transparency: Teams work in an environment where everyone is aware of the challenges faced by others. Regular face-to-face conversations between team members and project owners help avoid communication errors and information bottlenecks.
    - Reflection: Frequent reflection points are built into the framework to allow team members to review their progress. Project managers use the insights from these review meetings for future estimation and planning. As a result, projects can proceed more efficiently, adhering to budget and timeline.
    - Adaptation: Team members can redefine task priorities based on evolving client requirements. They decide which tasks to complete first and which to revisit later.

2. Values of Scrum for Project Teams

    Scrum teams adhere to five core values:

    - Commitment: Scrum team members commit to completing tasks and objectives within the given timeframe and are dedicated to continuous improvement to find the best solution.
    - Courage: Scrum teams demonstrate courage by asking open and challenging questions. They engage in honest and transparent discussions to arrive at the best solution.
    - Focus: For a given period, team members work from a product backlog containing tasks. They concentrate on selected tasks to deliver products within a limited timeframe.

- Openness: Scrum team members are open to new ideas and opportunities that promote individual learning and enhance the overall quality of the project.
- Respect: Team members respect project leaders, each other, and the Scrum process. This culture of respect fosters mutual collaboration and cooperation within the team.

1.4.2 Software tools

To effectively address the challenges and requirements of modern technological projects, a comprehensive understanding of various software tools and skills is essential. This includes fundamental programming capabilities, proficiency with development tools, expertise in managing virtual environments, and knowledge of electronics and robotics, as well as advanced skills in data analysis, machine learning, and artificial intelligence.

- Basic Programming Skills
  - Programming Languages: Python
  - Integrated Development Environments (IDEs): Visual Studio Code, IntelliJ IDEA
  - Version Control: Git, GitHub
- Working with Virtual Environments
  - Virtualization Platforms: VirtualBox, Docker
  - Environment Management: Anaconda, venv (Python)
  - Containerization Tools: Docker
- Data Analysis and Machine Learning
  - Data Analysis Tools: Pandas, NumPy, Excel, MATLAB
  - Machine Learning Libraries: TensorFlow, Scikit-learn
  - Data Visualization Tools: Matplotlib, Seaborn, Plotly, Tableau

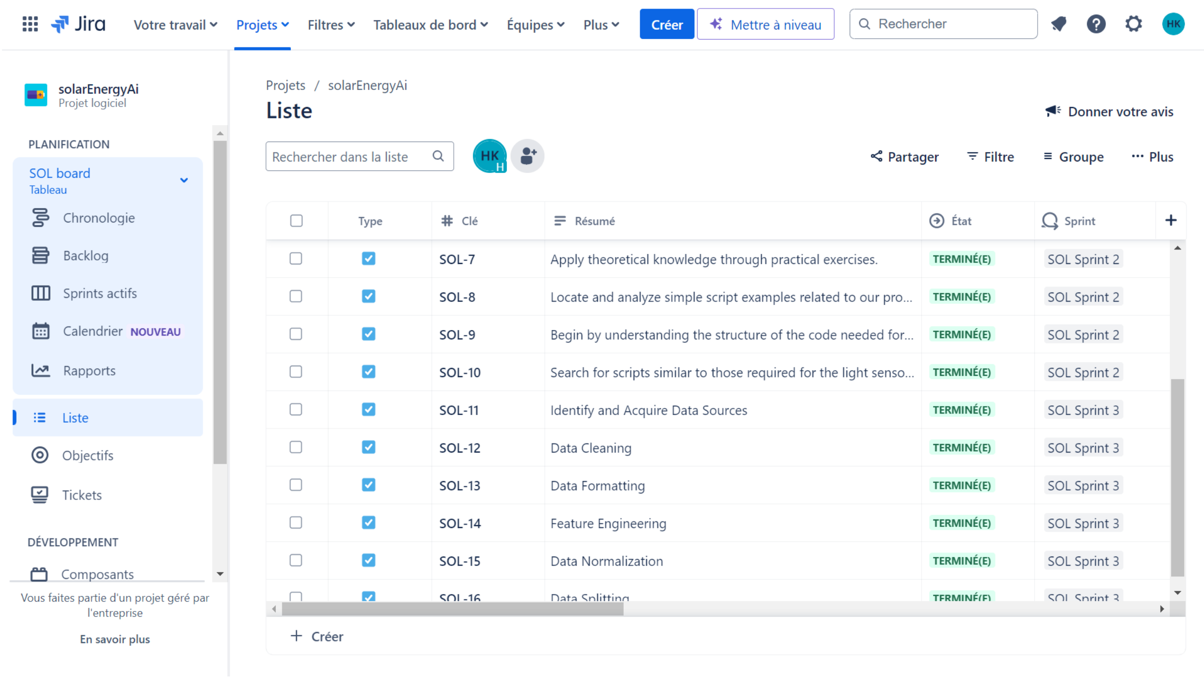


Figure 1.5.3: Sprint Management in Jira.

1.5 Task organisation and distribution

To structure and distribute tasks, we have used Jira as a project management tool to organize our project framework. See 1.5.3.
Jira is a software application developed by the Australian software company Atlassian that allows teams to track issues, manage projects, and automate workflows.

1.5.1 GANTT Chart

Project planning involves organizing and structuring the various tasks to be completed. Establishing a schedule means setting deadlines for each task and identifying the necessary milestones to achieve the project's objectives. Project management software is commonly used to automate task scheduling and time management.
A Gantt chart visually represents tasks chronologically based on sequencing constraints. This allows for a simple graphical representation to visualize the progress of each task within the project.
The figure illustrates the chronological sequence of the main steps we followed to successfully complete this project. See figure 1.5.4 and 1.5.5.

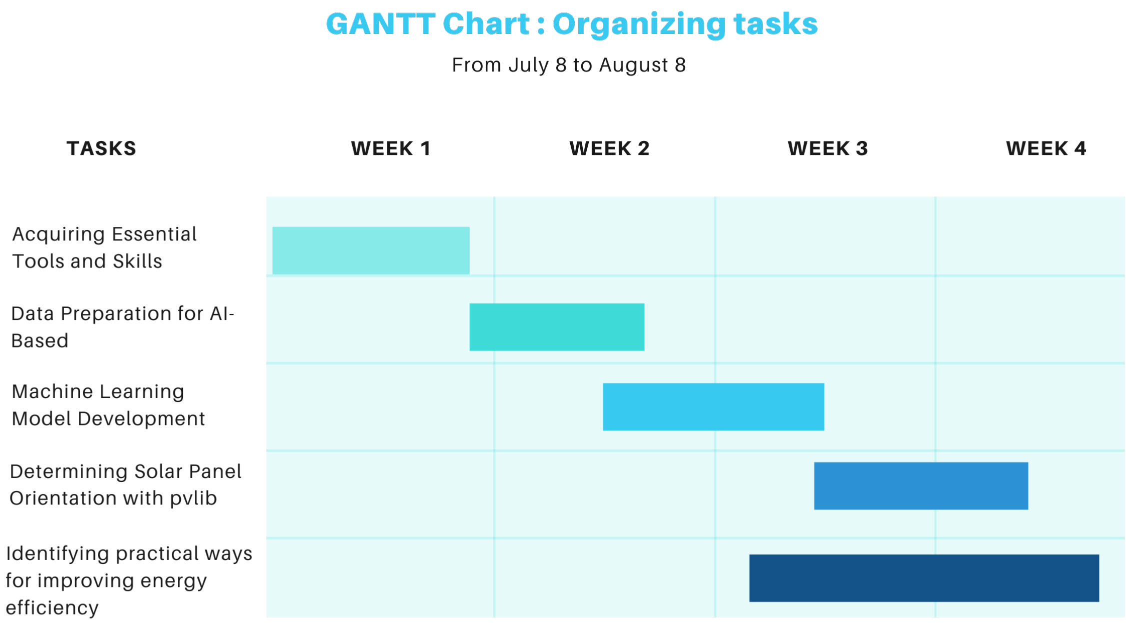


Figure 1.5.4: Task Organization for the First Four Weeks.

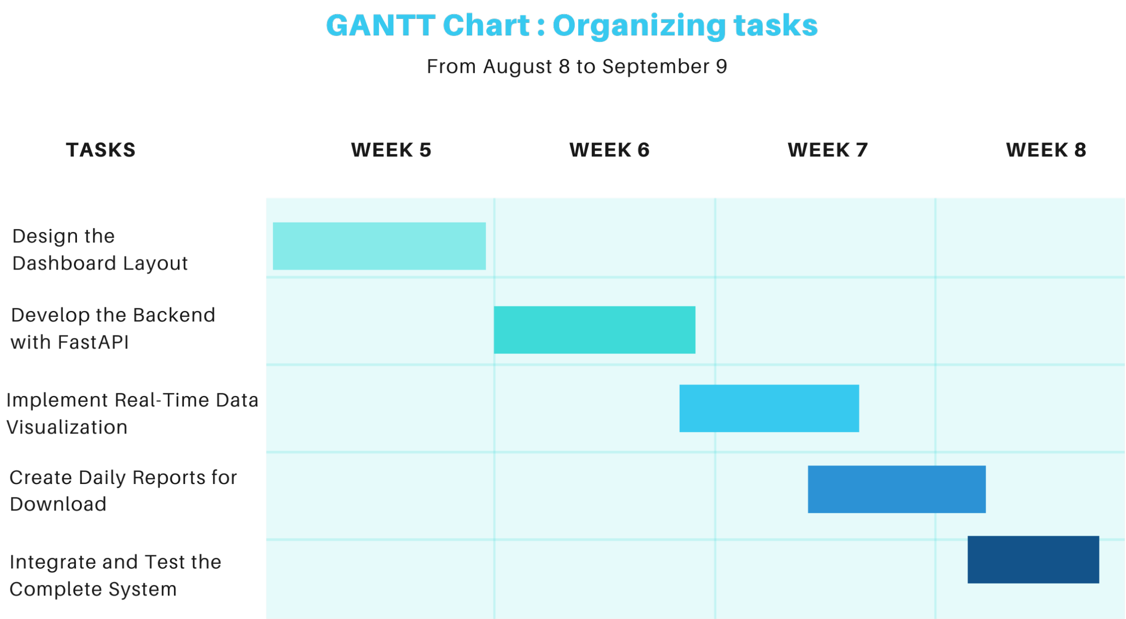


Figure 1.5.5: Task Organization for the Final Four Weeks.

1.6 Conclusion

This chapter provided an overview of UXV Center, detailing its activities and organizational structure. We reviewed the project's objectives and the essential tools and skills, including Scrum methodology and programming expertise. Additionally, we discussed the task organization, outlining the structured approach used to manage and execute the project. This chapter establishes a foundation for understanding the project's context, tools, and methodologies, as well as the strategic approach employed to achieve its goals.

Chapter 2

Overview of Solar Energy

2.1 Introduction

The increasing demand for energy, the continuous reduction in existing sources of fossil fuels and the growing concern regarding environment pollution, have pushed mankind to explore new technologies for the production of electrical energy using clean, renewable sources, such as solar energy, wind energy, etc. Among the non-conventional, renewable energy sources, solar energy affords great potential for conversion into electric power, able to ensure an important part of the electrical energy needs of the planet. The conversion of solar light into electrical energy represents one of the most promising and challenging energetic technologies, in continuous development, being clean, silent and reliable, with very low maintenance costs and minimal ecological impact. Solar energy is free, practically inexhaustible, and involves no polluting residues or greenhouse gases emissions. The conversion principle of solar light into electricity, called Photo-Voltaic or PV conversion, is not very new, but the efficiency improvement of the PV conversion equipment is still one of top priorities for many academic and industrial research groups all over the world.

2.2 Solar Energy and Its Various Applications

Solar energy, radiation from the Sun capable of producing heat, causing chemical reactions, or generating electricity. The total amount of solar energy incident on Earth is vastly in excess of the world's current and anticipated energy requirements. If suitably harnessed, this highly diffused source has the potential to satisfy all future energy needs. In the 21st century solar energy has become increasingly attractive as a renewable energy source because of its inexhaustible supply and its nonpolluting character, in stark contrast to the finite fossil fuels coal, petroleum, and natural gas.

As the solar power industry continues its positive influence on the energy market, we are beginning to see all of the prominent uses behind numerous solar energy applications. Besides, energy requires the sun's light and heat. By utilizing the sun's benefits to go solar,

we save money in the long run and take advantage of what renewable energy offers. As renewable energy trickles into our lives more with each passing day, we must understand the industrial applications of solar energy.
We will continue to see new applications of solar technologies to improve our environmental impact, reduce daily energy costs, and make our lives better overall.

- **Solar Heating for Buildings** :
  Space heating a building depends on whether or not a large building requires heat in its living spaces. For buildings with heat, solar heating needs to come from fans, ducts, radiators, or otherwise.

- **Solar-distillation** :
  The solar-distillation method requires ample sunlight to transform saline water into distilled water. Once the solar radiation turns into heat, it creates purified water for cooling purposes. Distilled water is normally expensive in other electrical avenues, but solar distillation makes this type of electrical energy more cost-effective.

- **Solar Drying for Agriculture** :
  Using solar energy to dry agricultural and animal products improve airflow and fruit quality, protecting sensitive agricultural products from harsh sunlight and preventing low moisture.

- **Solar Thermal Power Production** :
  Solar thermal power production is a method that transforms solar energy into electricity. It stores thermal energy by heating fluids through a turbine, producing steam to generate electricity. Ponds are one the most common bodies of water that store thermal energy for solar radiation.

- **Solar Electric Power Generation** :
  Photovoltaic (PV) cells generate electricity through direct sunlight. There are various electrical benefits to using solar electric power generation, such as reliability, low maintenance costs, durability, and eco-friendly. Solar electric power generation is

most beneficial for irrigation, commercial-grid power systems, public transportation, and more.

2.3 From Cell to Photovoltaic Panel

A solar cell, also known as a photovoltaic cell, is the fundamental component of a photovoltaic system designed to convert sunlight into electrical energy through the photovoltaic effect. It operates by generating a small amount of electricity when exposed to sunlight, thanks to its construction from semiconductor materials like silicon.

Each solar cell comprises two layers of semiconductor material: a positive (p-type) and a negative (n-type) layer, which together create an electric field. While a single solar cell typically produces a voltage of around 0.5 to 0.6 volts and a modest current, it is insufficient for substantial power needs on its own. Therefore, multiple solar cells are connected together in a grid pattern within a panel to generate usable amounts of electrical power.

These cells are generally flat and come in various shapes, including round and rectangular, making them adaptable for integration into solar panels.

A photovoltaic (PV) panel is an integrated assembly of multiple solar cells connected in series and/or parallel to create a larger unit capable of generating substantial amounts of electricity. Unlike a single solar cell, which produces a modest voltage and current, a PV panel is designed to provide higher voltage and current, typically ranging from 250 to 400 watts, depending on its size and the efficiency of its constituent cells. In addition to the solar cells, a PV panel includes essential components such as a protective cover (usually made of glass or plastic), a backing material, and sometimes a supporting frame for mounting. It also features electrical connections for wiring, allowing it to be integrated into solar energy systems. Physically, PV panels are larger and rectangular, commonly installed on rooftops or mounted on stands to optimize sunlight capture. In summary, while a solar cell is the basic unit that directly converts sunlight into electricity, a PV panel is a collection of these cells designed to produce a more practical and substantial amount of electrical power, ready for use in solar energy systems. See figure 2.3.1

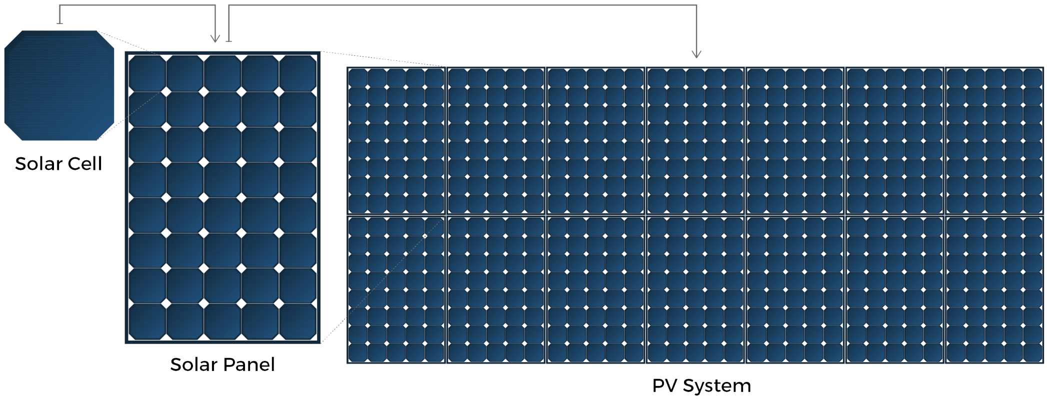


Figure 2.3.1: From a solar cell.

2.4 The Different Materials Used in Photovoltaic Panel Manufacturing

PV materials and devices convert sunlight into electrical energy.
A single PV device is known as a cell. An individual PV cell is usually small, typically producing about 1 or 2 watts of power. These cells are made of different semiconductor materials and are often less than the thickness of four human hairs. In order to withstand the outdoors for many years, cells are sandwiched between protective materials in a combination of glass and/or plastics.
To boost the power output of PV cells, they are connected together in chains to form larger units known as modules or panels. Modules can be used individually, or several can be connected to form arrays. One or more arrays is then connected to the electrical grid as part of a complete PV system. Because of this modular structure, PV systems can be built to meet almost any electric power need, small or large.
PV modules and arrays are just one part of a PV system. Systems also include mounting structures that point panels toward the sun, along with the components that take the direct-current electricity produced by modules and convert it to the alternating-current electricity used to power all of the appliances in your home.

Silicon is one of the most important materials used in solar panels, making up the semiconductors that create electricity from solar energy. However, the materials used to manufacture the cells for solar panels are only one part of the solar panel itself. The manufacturing process combines six components to create a functioning solar panel. These parts include silicon solar cells, a metal frame, a glass sheet, standard 12V wire, and bus wire.
Here are the common parts of a solar panel explained :

- **Silicon solar cells :**
  Silicon solar cells convert the Sun's light into electricity using the photovoltaic effect. Soldered together in a matrix-like structure between the glass panels, silicon cells interact with the thin glass wafer sheet and create an electric charge.

- **Metal frame (typically aluminum) :**
  A solar panel's metal frame is useful for many reasons; protecting against inclement weather conditions or otherwise dangerous scenarios and helping mount the solar panel at the desired angle.

- **Glass sheet :**
  The glass casing sheet is usually 6-7 millimeters thick, and although it is thin, it plays a significant role in protecting the silicon solar cells inside.

- **Standard 12V wire :**
  A 12V wire helps regulate the amount of energy being transferred into your inverter, aiding with the sustainability and efficiency of the solar module.

- **Bus wires :**
  Bus wires are used to connect the silicon solar cells in parallel. Bus wires are covered in a thin layer for easy soldering and are thick enough to carry electrical currents.

2.5 The Principle of Photovoltaic Conversion

Direct use of solar energy can be performed in essentially two different ways: (1) the transformation of sunlight directly into electricity in semiconducting devices that are more popularly known as solar cells; and (2) the collection of heat in solar collectors. The transformation of solar radiation into electrical current is referred to as **"photovoltaic energy conversion"** (PV), and this is the meaning of the word "photovoltaic energy conversion."
The photovoltaic effect is responsible for this phenomenon. The phenomena that can cause a potential difference to occur at the interface of two materials that are not identical is referred to as the **"photovoltaic effect"**, and it is described using the word "photovoltaic effect." As a result, the entire field that studies the conversion of solar energy into electricity is referred to as "photovoltaics," and its acronym stands for "photovoltaic electrics." The term "photovoltaics" comes from the combination of the Greek word for light ("photo") with the Italian name of an early electrical researcher, Alessandro Volta (1745–1827), which is shortened to "Volt." Since most people do not know what the word "photovoltaics" means, the term "solar electricity" has become the most frequent way of referring to PV solar energy.

Three primary processes are necessary for the photovoltaic effect to take place: (1) the generation of charge carriers as a result of photon absorption by the materials comprising a junction; (2) the subsequent separation of the photo-generated charge carriers within the junction; and (3) the collection of the photo-generated charge carriers at the terminals of the junction.

2.5.1 History of photovoltaic effect

The discovery of the photovoltaic phenomenon is attributed to a French physicist named Alexandre Edmond Becquerel in the year 1839. During the course of his experiments using metal electrodes and electrolyte, he noticed that the conductivity increased as the amount of light increased. Willoughby Smith made the discovery that selenium possesses photovoltaic properties in the year 1873.

Actually, when a material absorbs light at a frequency over a threshold frequency that varies with the substance, electrons are released. This phenomenon is called **the photoelectric effect**, and it is closely related to the photovoltaic effect. Taking into account the fact that light is presumed to be made up of individual energy quanta (photons), Albert Einstein was able to explain this phenomenon in 1905. This type of photon's energy can be calculated :

$$E = h\nu$$

where h is Planck's constant and ν is the frequency of the light.

In 1921, Albert Einstein published the paper on the photoelectric effect for which he received his sole Nobel Prize. After another decade, the first pure semiconductor was developed in 1931. Solar cells were first used for space applications in the 1950s. In a short amount of time, Hoffman Electronics was able to surpass the previous record for solar cell efficiency, achieving highs of 10 and 14% in the years 1959 and 1960, respectively. The earliest solar cells, which had an efficiency of roughly 8%, were invented in 1957. A short time after that, the first PV cell made of amorphous silicon was developed, and the capacity of PV systems reached 500 kW. This amount continued to rise, reaching its highest point of 21.3 MW in the year 1983. A high-concentrating photovoltaic (PV) facility with a capacity of 175 kilowatts (kW) was finally constructed in the state of Arizona in the United States in the year 2002. A new PV technology efficiency record of 40% was set 4 years later. In 2012, when the global PV capacity hit 100 GW, production prices dropped drastically to $1.25 per watt. The first solar-powered aircraft completed a global flight in 2016 . The global PV power potential is shown in Figure 1 2.5.2. Solar photovoltaic generation will increase by 23 percent, from 156 GWh in 2015 to 821 GWh in 2020, making it the fastest-growing renewable energy source after wind and ahead of hydropower. PV capacity additions experienced an exceptional rise (a record of 134 GW) in China, the US, and Vietnam. Unquestionably, solar PV is moving toward becoming the most affordable choice for producing power globally, and in the years to come, it is anticipated to draw significant investment.

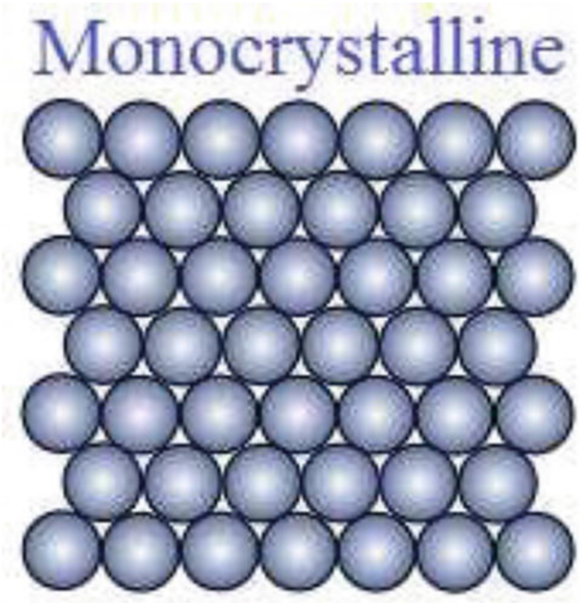


Figure 2.5.2: Monocrystalline structure.

2.5.2 Photovoltaic effect

The photovoltaic effect is a process that generates voltage or electric current in a photovoltaic cell when it is exposed to sunlight. It is this effect that makes solar panels useful, as it is how the cells within the panel convert sunlight to electrical energy.

Process

The photovoltaic effect occurs in solar cells. These solar cells are composed of two different types of semiconductors - a p-type and an n-type - that are joined together to create a p-n junction. By joining these two types of semiconductors, an electric field is formed in the region of the junction as electrons move to the positive p-side and holes move to the negative n-side. This field causes negatively charged particles to move in one direction and positively charged particles in the other direction.
Light is composed of photons, which are simply small bundles of electromagnetic radiation or energy. These photons can be absorbed by a photovoltaic cell - the type of cell that composes solar panels. When light of a suitable wavelength is incident on these cells, energy from the photon is transferred to an atom of the semiconducting material in the p-n junction. Specifically, the energy is transferred to

the electrons in the material. This causes the electrons to jump to a higher energy state known as the conduction band. This leaves behind a "hole" in the valence band that the electron jumped up from. This movement of the electron as a result of added energy creates two charge carriers, an electron-hole pair.

When unexcited, electrons hold the semiconducting material together by forming bonds with surrounding atoms, and thus they cannot move. However in their excited state in the conduction band, these electrons are free to move through the material. Because of the electric field that exists as a result of the p-n junction, electrons and holes move in the opposite direction as expected. Instead of being attracted to the p-side, the freed electron tends to move to the n-side. This motion of the electron creates an electric current in the cell. Once the electron moves, there's a "hole" that is left. This hole can also move, but in the opposite direction to the p-side. It is this process which creates a current in the cell. A diagram of this process can be seen in Figure 2.5.3.

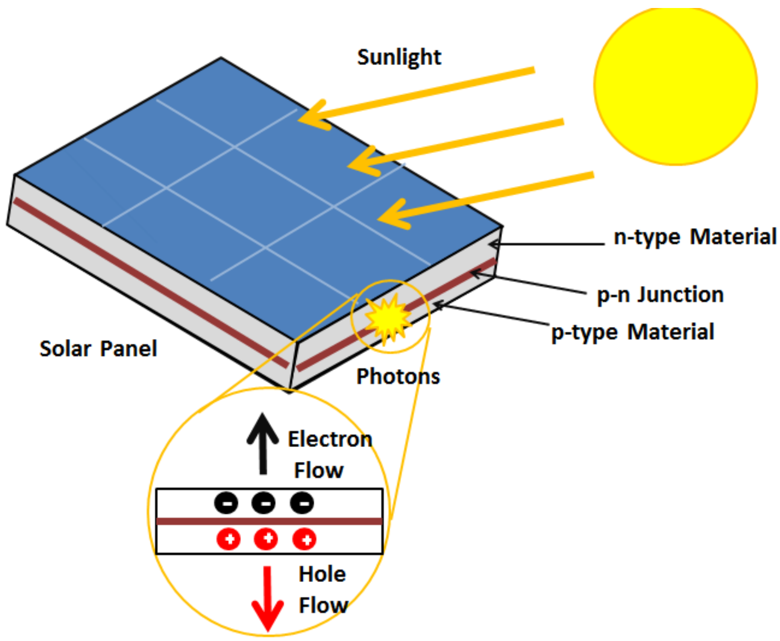


Figure 2.5.3: A diagram showing the photovoltaic effect.

2.6 Factors Influencing Solar Panel Efficiency (orientation)

Solar panels, or photovoltaic (PV) panels, are designed to convert sunlight into electrical energy. Their efficiency—the proportion of solar energy converted into usable electrical power—can vary significantly based on several factors. Understanding these factors is crucial for optimizing solar panel performance, which can significantly influence the effectiveness of solar energy systems. Key factors affecting solar panel efficiency include the quality of the solar cells, environmental conditions, and the orientation and angle of the panels.
Let's explore the factors that can make all the difference in ensuring solar panel efficiency :

- **Geographic Location** : The effectiveness of solar panel installations is significantly influenced by geographic location. Regions closer to the equator receive more direct sunlight throughout the year, enhancing solar energy production. However, local weather conditions such as precipitation, pollution, and fog can impact efficiency. Despite these factors, solar panels can still generate power in cloudy conditions.

- **Time of Day** : How to maximize the efficiency of your home solar panel systems? You may need to understand that peak sun hours are crucial. Unlike daylight hours, peak sun hours refer to the time frame when solar irradiance hits a power density of 1,000 watts per square meter. Typically occurring between 11 am and 4 pm. These hours coincide with the sun's highest position in the sky, ensuring optimal solar radiation for your panels.

- **Seasonal Changes** : The efficiency of solar panels is also influenced by seasonal variations. The angle and intensity of the sun's rays change throughout the year, which can affect the amount of solar energy your panels receive. Additionally, while high temperatures are often thought to improve efficiency, they can actually decrease it. Snowfall can have varying effects; it might obstruct panel exposure or, conversely, enhance performance depending on how it impacts sunlight access.

- **Obstructions and Shading** : To maintain optimal efficiency in your solar panel system, it's crucial to ensure that no objects or trees cast shadows on your panels. Shading can significantly reduce the system's performance, potentially cutting efficiency by more than half and negatively affecting your return on investment. To prevent this, regularly trim branches and remove any obstructions.

- **Roof Pitch and Orientation**: The orientation and pitch of your roof can affect the efficiency of your solar panels. Panels facing east or west typically produce about 20% less electricity compared to those facing south. However, they can still be cost-effective. To meet your energy needs, you might need to install additional panels. In the northern hemisphere, north-facing roofs are generally less effective for solar energy production. Consider alternative setups such as ground-mounted systems or carport installations to maximize solar energy collection.

2.7 Solar Trackers: An Optimal Solution to Maximize Solar Panel Performance

2.7.1 Introduction

As the demand for renewable energy solutions continues to grow, optimizing solar panel efficiency becomes increasingly crucial. As discussed in the previous chapter, one of the most significant factors affecting solar energy capture is the orientation of the panels. Since the sun's trajectory shifts throughout the day and varies with the seasons, employing innovative strategies to fully harness its energy potential is essential.
This is where solar trackers come into play. These advanced systems automatically adjust the orientation of solar panels in real-time to follow the sun's path across the sky. By keeping the panel surfaces perpendicular to the solar rays, solar trackers maximize the capture of solar radiation. This continuous alignment with the sun enhances

energy production and boosts the overall efficiency of solar power systems.
Beyond optimizing orientation, solar trackers address challenges posed by seasonal variations and daily fluctuations in sunlight. They adapt to changing solar angles, ensuring that panels are always positioned to capture the maximum amount of solar energy throughout the year. This dynamic adjustment not only increases energy yield but also provides a more cost-effective and sustainable energy solution.

2.7.2 Solar Tracking and PV Panel Efficiency

Compared to a fixed panel, a mobile PV panel driven by a solar tracker is kept under the best possible insolation for all positions of the Sun, as the light falls close to the geometric normal incidence angle. Automatic solar tracking systems (using light intensity sensing) may boost consistently the conversion efficiency of a PV panel, thus in this way deriving more energy from the sun.
Technical reports in the USA have shown solar tracking to be particularly effective in summer, when the increases in output energy may reach over 50%, while in autumn they may be higher than 20%, depending on the technology used.
Solar tracking systems are of several types and can be classified according to several criteria. A first classification can be made depending on the number of rotation axes. Thus we can distinguish solar tracking systems with a rotation axis, respectively with two rotation axes. Since solar tracking implies moving parts and control systems that tend to be expensive, single-axis tracking systems seem to be the best solution for small PV power plants. Single axis trackers will usually have a manual elevation (axis tilt) adjustment on the second axis which is adjusted at regular intervals throughout the year.
A single-axis solar tracking system uses a tilted PV panel mount, Fig 2.7.4, and a single electric motor to move the panel on an approximate trajectory relative to the Sun's position.

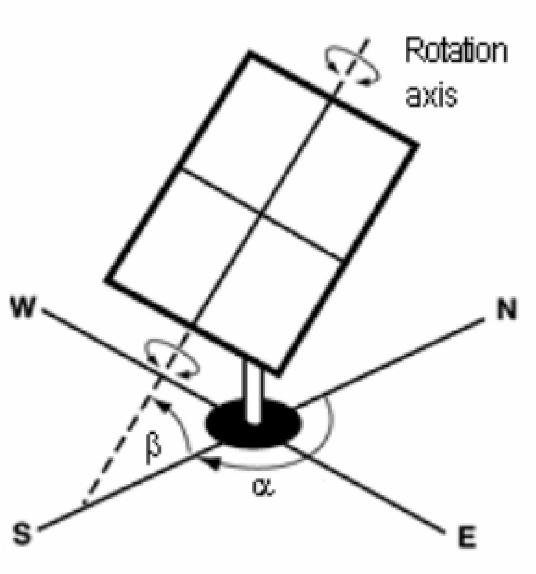


Figure 2.7.4: Principle of the single-axis solar tracking system .

Another classification of solar tracking systems can be made depending on the orientation type. According to this criterion, we can identify solar tracking systems that orient the PV panels based on a previously computed sun trajectory, in comparison with panels with an on-line orientation system that reacts to the instantaneous solar light radiation. The later solution is more efficient and it was chosen for the solar tracking system proposed in this paper. Another criterion for the solar tracker classification refers to its activity type. According to this criterion, we can distinguish active or passive solar trackers.

How Solar Trackers Work ?

The fundamental principle of solar trackers is based on accurately detecting the sun's position using sensors such as light sensors or GPS systems. These sensors provide real-time data, which is used to adjust the angle and orientation of the solar panels, ensuring they remain perpendicular to the incident solar rays.
Solar trackers utilize various technologies to determine the sun's position. Light sensors or photoresistors measure sunlight intensity and adjust the panels accordingly. Advanced systems may use GPS and astronomical algorithms to calculate the sun's precise location throughout the day and year. In addition, modern solar trackers

can incorporate artificial intelligence (AI) to enhance their performance. AI algorithms analyze historical weather data, solar irradiance patterns, and system performance metrics to make predictive adjustments and optimize panel orientation more effectively.

Benefits of Solar Trackers

The advantages of using solar trackers are significant. By dynamically adjusting the position of the solar panels, trackers can enhance the efficiency of photovoltaic systems, often improving energy yield by 20% to 40% compared to fixed installations. This boost in performance is due to the trackers' ability to maintain panels in an optimal orientation relative to the sun, maximizing solar energy absorption throughout the day.
Additionally, solar trackers help to make more efficient use of available space by allowing a higher density of solar panel installations. By optimizing each panel's angle relative to the sun, trackers ensure that every panel operates at peak performance, which is particularly valuable in space-constrained environments with high energy demands. The integration of AI further refines this optimization, allowing solar trackers to anticipate changes in weather conditions and adjust panel positions proactively, thereby maximizing overall system efficiency.

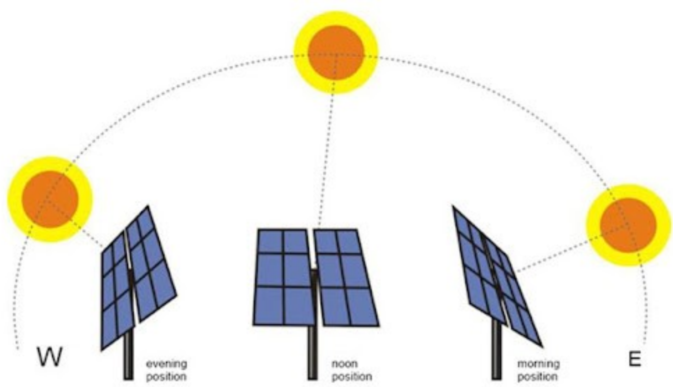


Figure 2.7.5: Solar tracker.

2.7.3 Design of the Proposed Solar Tracking System

The proposed solar tracking system should satisfy certain technical requirements specific to the studied application, as follows:

- minimum energy consumption, for the maximization of global efficiency of the installation and optimum performance-cost ratio.
- reliability in operation, under different perturbation conditions (wind, dust, rain, important temperature variations).
- simplicity of movement solution (motor, gears, sensors), to diminish the cost and to increase the viability
- possibility of system integration in a monitoring and control centralized structure, which means a digital control solution.

2.8 Conclusion

This chapter has outlined the core principles and applications of solar energy. It covered the fundamental workings of solar energy, the materials used in solar panel manufacturing, and the factors affecting panel efficiency, such as orientation. Additionally, it discussed the role of solar trackers in enhancing panel performance. These insights collectively highlight the advancements and ongoing innovations in solar technology, emphasizing their impact on improving renewable energy systems.

Chapter 3

Machine Learning Essentials

3.1 Introduction

While humans learn from experience, can computers do the same? The answer is "yes", and machine learning is what we need.
In this chapter, we will cover the essentials of machine learning, starting with its definition and significance. We will explore the three main types of machine learning—supervised, unsupervised, and reinforcement learning—and discuss different types of problems such as regression and classification. Additionally, we will outline the key steps to solving a machine learning problem, including dataset preparation, model selection, training, and evaluation using various performance metrics. This overview will provide a solid foundation for understanding and implementing machine learning in our project.

3.2 Foundations of Machine Learning

3.2.1 Definition of Machine Learning

Machine learning is the technique that improves system performance by learning from experience via computational methods. In computer systems, experience exists in the form of data, and the main task of machine learning is to develop learning algorithms that build models from data.By feeding the learning algorithm with experience data, we obtain a model that can make predictions on new observations.If we consider computer science as the subject of algorithms, then machine learning is the subject of learning algorithms.

3.2.2 Importance of Machine Learning

Machine learning algorithms have proven their significance across a diverse range of applications, demonstrating their versatility and impact. In text and document classification, they are used for tasks such as spam detection, enhancing our ability to manage and filter information effectively.In natural language processing, machine learning techniques contribute to morphological analysis, part-of-speech tagging, statistical parsing, and named-entity recognition, facilitating more sophisticated and accurate interactions with hu-

man language. The field also extends to speech recognition and synthesis, as well as speaker verification, which are essential for developing advanced voice-activated systems.
Additionally, optical character recognition (OCR) leverages machine learning for converting different types of documents into editable data.In computational biology, these algorithms aid in predicting protein functions and structures, offering critical insights for research and development. Machine learning also plays a crucial role in computer vision tasks, such as image recognition and face detection, which are integral to various technologies. Moreover, they are employed in fraud detection systems for credit card and telephone transactions, as well as in network intrusion prevention.The application extends to gaming, where machine learning enhances strategies in chess and backgammon, and unassisted vehicle control, which is pivotal for advancements in robotics and navigation. In the medical field, machine learning assists in diagnosing diseases with greater accuracy.
Lastly, recommendation systems, search engines, and information extraction systems rely on these algorithms to provide personalized and relevant content to users, underscoring their importance in both everyday applications and complex problem-solving scenarios.

3.3 The Different Types of Machine Learning

We next briefly describe common machine learning scenarios. These scenarios differ in the types of training data available to the learner, the order and method by which training data is received and the test data used to evaluate the learning algorithm.

3.3.1 Supervised Learning

Supervised learning involves training a model using a labeled dataset, where each input is associated with a specific output. The goal is to learn a mapping from inputs to outputs that can be applied to new, unseen data. This type of learning is used for tasks where historical data with known outcomes is available.

3.3.2 Unsupervised Learning

Unsupervised learning is used when the data lacks labeled responses. The aim is to uncover hidden patterns or structures within the data. This type of learning is useful for exploratory data analysis and identifying relationships without predefined categories

3.3.3 Reinforcement Learning

Reinforcement learning involves training an agent to make decisions through trial and error, receiving feedback in the form of rewards or penalties. The objective is to learn a strategy that maximizes cumulative rewards over time. This approach is well-suited for problems involving sequential decision-making and long-term planning.

3.4 Types of Machine Learning Problems

Machine learning problems can be broadly categorized based on the nature of the task and the type of data involved. The main types include regression problems, classification problems, and a few others that address different aspects of machine learning.Some major classes of learning problems are:

3.4.1 Regression problems

Predict a real value for each item. Examples of regression include prediction of stock values or variations of economic variables. In this problem, then penalty for an incorrect prediction depends on the magnitude of the difference between the true and predicted values, in contrast with the classification problem, where there is typically no notion of closeness between various categories.

3.4.2 Classification problems

Assign a category to each item. For example, document classification may assign items with categories such as politics, business, sports, or weather while image classification may assign items with categories such as landscape, portrait, or animal. The number of categories in such tasks is often relatively small, but can be large in some difficult tasks and even unbounded as in OCR, text classification, or speech recognition.

3.4.3 Clustering problems

Partition items into homogeneous regions. Clustering is often performed to analyze very large data sets. For example, in the context of social network analysis, clustering algorithms attempt to identify "communities" within large groups of people.

3.5 Fundemental Steps for Solving a Machine Learning Problem

Solving a machine learning problem involves a series of systematic steps to develop, train, and evaluate a model that can make accurate predictions or classifications. Here are the fundamental steps typically followed:

3.5.1 Define the Problem

- <u>Objective</u>: Clearly articulate the problem you are trying to solve and determine the goals of the machine learning project. This includes understanding what you need the model to predict or classify and how the results will be used.

- <u>Questions to Consider</u>: What is the target variable? What type of problem is it (regression, classification, etc.)? What are the success criteria?

3.5.2 Collect and Prepare Data

- <u>Data Collection</u>: Collect the relevant data required for the problem by extracting information from databases, web scraping, using APIs, or through surveys and experiments. Additionally, consider leveraging platforms such as Kaggle, UCI Machine Learning Repository, and Google Dataset Search to access and obtain datasets.

- Data Cleaning: Process and clean the data to handle missing values, remove duplicates, and correct errors. This may also involve normalizing or standardizing data.

- Feature Engineering: Create new features from existing data to improve the model's performance. This may include selecting important features, transforming variables, or creating interaction terms.

3.5.3 explore and analyze data

- Exploratory Data Analysis (EDA): Use statistical summaries and visualizations to understand the data's distribution, relationships, and patterns. This helps in identifying trends, correlations, and potential issues.

- Descriptive Statistics: Calculate measures such as mean, median, variance, and correlation to summarize the data.

- Visualization: Create plots (e.g., histograms, scatter plots) to visualize data distributions and relationships between variables.

3.5.4 Selecting appropriate model based on the problem

- Model Selection: Choose appropriate machine learning algorithms based on the problem type and data characteristics. Common algorithms include linear regression for regression problems, decision trees for classification, and clustering algorithms for unsupervised learning.

- Algorithm Choice: Consider factors such as the complexity of the model, interpret-ability, and computational resources.

3.5.5 Split the Data

- Training and Test Sets: Divide the data into training and testing subsets. The training set is used to train the model, while the test set is used to evaluate its performance.

- Validation Set: Optionally, use a validation set (or employ cross-validation) to tune hyper parameters and prevent over-fitting.

3.5.6 Train the model

- Model Training: Apply the selected algorithm to the training data to fit the model. This involves adjusting the model parameters to minimize errors and optimize performance.
- Hyperparameter Tuning: Adjust hyperparameters (e.g., learning rate, number of trees) to improve the model's performance.

3.5.7 Evaluate the Model:

- Performance Metrics: Assess the model's performance using appropriate metrics. For regression, common metrics include Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). For classification, metrics include accuracy, precision, recall, F1 score, and the ROC-AUC.
- Cross-Validation: Use cross-validation to ensure that the model generalizes well to unseen data and to mitigate over-fitting.

3.5.8 Conclusion

In this chapter, we have explored the fundamental concepts and methodologies essential for tackling machine learning problems. We began by defining the core principles of machine learning, followed by an overview of different types of machine learning, including supervised, unsupervised, and reinforcement learning. We then delved into specific machine learning problems, such as regression and classification, and outlined the fundamental steps involved in solving these problems—from data collection and preparation to model training and evaluation.

Understanding these concepts is crucial for effectively applying machine learning techniques to real-world challenges. The insights gained from this chapter will serve as a foundation for developing and deploying machine learning models in subsequent tasks.

In the next chapter, we will apply the principles and methods discussed here to explore AI solutions for optimizing panel orientation. We will examine how machine learning can be utilized to enhance the efficiency and performance of panel systems, incorporating the methodologies and techniques from this chapter to address the specific needs of our project

Chapter 4

AI Solutions for Optimizing Panel Orientation

4.1 Introduction

Our UGV is designed to operate using solar energy, utilizing a solar panel mounted on its surface. The goal is not only to power the vehicle using solar radiation but, more importantly, to optimize this energy to maximize the vehicle's operating time. To achieve this, it is crucial to continuously adjust the orientation of the solar panel so that it can track the sun's path throughout the day, thereby improving the vehicle's energy efficiency. A variety of techniques can be employed to address the challenge of optimizing solar energy capture. Among these, artificial intelligence stands out as a powerful tool capable of providing optimal solutions to a wide range of complex problems. In our case, we aim to develop a system that uses AI to dynamically determine the optimal orientation angle of the photovoltaic panel. This system will continuously adjust the panel's position to ensure it is always facing the sun at the most effective angle.

4.2 Optimizing panel orientation using a machine learning model

In machine learning, the goal is to enable the machine to make predictions based on existing data compiled into a dataset. In our case, we aim to predict the position of the sun based on the vehicle's geographic location and the time during which it operates.

4.2.1 Dataset preparation

In our project, achieving the optimal orientation of the solar panel requires identifying and setting two distinct angles that constitute the targets of our machine learning model :

- **The elevation angle** which determines how high or low the panel is tilted relative to the sun's position in the sky. Adjusting this angle is crucial for capturing the maximum amount of solar radiation as the sun moves from east to west throughout the day.

- **The azimuth angle** represents the angle of the panel relative to the north, indicating its direction along the horizontal plane. This angle helps ensure that the panel is oriented in the correct direction to face the sun throughout the day.

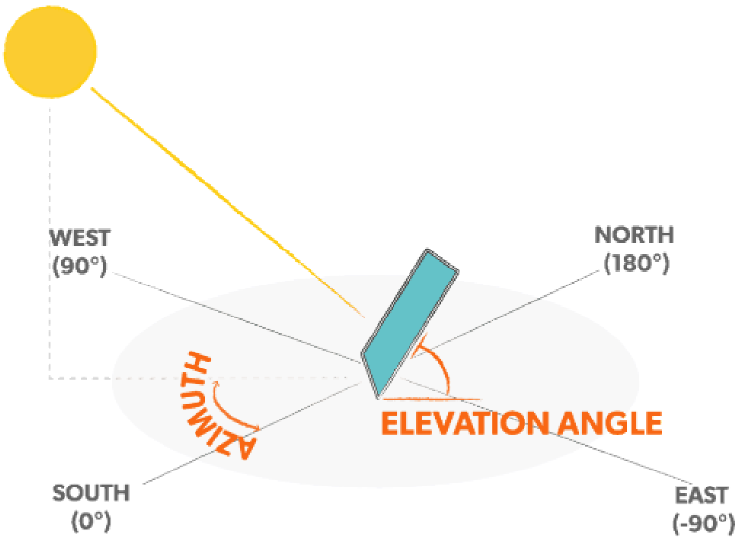


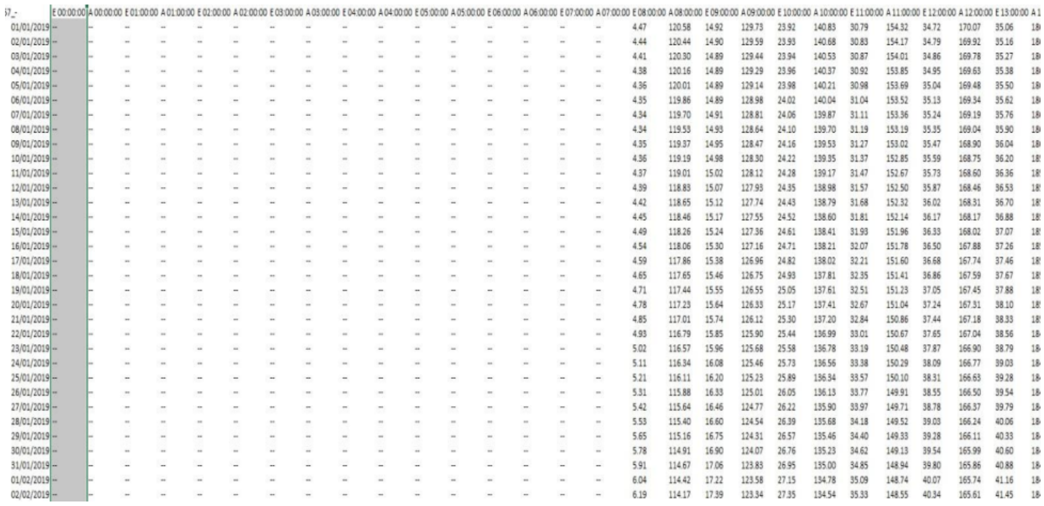
Figure 4.2.1: Azimuth and Tilt angles.

Determining these two angles requires taking several factors into account, including the **vehicle's geographic location** (latitude and longitude), **the date**, and **the time**. These factors are going to be the features of our machine learning model.

a) Gathering the data

To begin our project, we first conducted an in-depth search for reliable sources from which we could download a dataset. This dataset will serve as the foundation for our work, as it will be used to train our machine learning model. The goal is to find relevant data of historical sun's position that is well-suited to our problem, in order to prepare and utilize it effectively for the development of the model. The original version that we processed was from the website:

`https://www.sunearthtools.com/dp/tools/pos_sun.php#annual`



57_-	E 00:00:00	A 00:00:00	E 01:00:00	A 01:00:00	E 02:00:00	A 02:00:00	E 03:00:00	A 03:00:00	E 04:00:00	A 04:00:00	E 05:00:00	A 05:00:00	E 06:00:00	A 06:00:00	E 07:00:00	A 07:00:00	E 08:00:00	A 08:00:00	E 09:00:00	A 09:00:00	E 10:00:00	A 10:00:00	E 11:00:00	A 11:00:00	E 12:00:00	A 12:00:00	E 13:00:00	A 1
01/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.47	120.58	14.92	129.73	23.92	140.83	30.79	154.32	34.72	170.07	35.06	18
02/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.44	120.44	14.90	129.59	23.93	140.68	30.83	154.17	34.79	169.92	35.16	18
03/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.41	120.30	14.89	129.44	23.94	140.53	30.87	154.01	34.86	169.78	35.27	18
04/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.38	120.16	14.89	129.29	23.96	140.37	30.92	153.85	34.95	169.63	35.38	18
05/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.36	120.01	14.89	129.14	23.98	140.21	30.98	153.69	35.04	169.48	35.50	18
06/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.35	119.86	14.89	128.98	24.02	140.04	31.04	153.52	35.13	169.34	35.62	18
07/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.34	119.70	14.91	128.81	24.06	139.87	31.11	153.36	35.24	169.19	35.76	18
08/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.34	119.53	14.93	128.64	24.10	139.70	31.19	153.19	35.35	169.04	35.90	18
09/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.35	119.37	14.95	128.47	24.16	139.53	31.27	153.02	35.47	168.90	36.04	18
10/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.36	119.19	14.98	128.30	24.22	139.35	31.37	152.85	35.59	168.75	36.20	18
11/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.37	119.01	15.02	128.12	24.28	139.17	31.47	152.67	35.73	168.60	36.36	18
12/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.39	118.83	15.07	127.93	24.35	138.98	31.57	152.50	35.87	168.46	36.53	18
13/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.42	118.65	15.12	127.74	24.43	138.79	31.68	152.32	36.02	168.31	36.70	18
14/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.45	118.46	15.17	127.55	24.52	138.60	31.81	152.14	36.17	168.17	36.88	18
15/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.49	118.26	15.24	127.36	24.61	138.41	31.93	151.96	36.33	168.02	37.07	18
16/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.54	118.06	15.30	127.16	24.71	138.21	32.07	151.78	36.50	167.88	37.26	18
17/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.59	117.86	15.38	126.96	24.82	138.02	32.21	151.60	36.68	167.74	37.46	18
18/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.65	117.65	15.46	126.75	24.93	137.81	32.35	151.41	36.86	167.59	37.67	18
19/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.71	117.44	15.55	126.55	25.05	137.61	32.51	151.23	37.05	167.45	37.88	18
20/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.78	117.23	15.64	126.33	25.17	137.41	32.67	151.04	37.24	167.31	38.10	18
21/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.85	117.01	15.74	126.12	25.30	137.20	32.84	150.86	37.44	167.18	38.33	18
22/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	4.93	116.79	15.85	125.90	25.44	136.99	33.01	150.67	37.65	167.04	38.56	18
23/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.02	116.57	15.96	125.68	25.58	136.78	33.19	150.48	37.87	166.90	38.79	18
24/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.11	116.34	16.08	125.46	25.73	136.56	33.38	150.29	38.09	166.77	39.03	18
25/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.21	116.11	16.20	125.23	25.89	136.34	33.57	150.10	38.31	166.63	39.28	18
26/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.31	115.88	16.33	125.01	26.05	136.13	33.77	149.91	38.55	166.50	39.54	18
27/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.42	115.64	16.46	124.77	26.22	135.90	33.97	149.71	38.78	166.37	39.79	18
28/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.53	115.40	16.60	124.54	26.39	135.68	34.18	149.52	39.03	166.24	40.06	18
29/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.65	115.16	16.75	124.31	26.57	135.46	34.40	149.33	39.28	166.11	40.33	18
30/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.78	114.91	16.90	124.07	26.76	135.23	34.62	149.13	39.54	165.99	40.60	18
31/01/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	5.91	114.67	17.06	123.83	26.95	135.00	34.85	148.94	39.80	165.86	40.88	18
01/02/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	6.04	114.42	17.22	123.58	27.15	134.78	35.09	148.74	40.07	165.74	41.16	18
02/02/2019	--	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	6.19	114.17	17.39	123.34	27.35	134.54	35.33	148.55	40.34	165.61	41.45	18

Figure 4.2.2: dataset before preprocessing.

b) Pre-processing the data

After uploading the initial version of our dataset, we began processing it by removing inconsistencies and addressing any missing values that could affect the accuracy of our results. This resulted in a clean version of the dataset, containing only the essential and relevant information.

A	B	C	D	E	F	G
MONTH	DAY	HEURE	PANEL_TILTE	PANEL_ORIEN	SUNRISE	SUNSET
1	1	00:00	0	0	08:29	18:38
1	1	01:00	0	0	08:29	18:38
1	1	02:00	0	0	08:29	18:38
1	1	03:00	0	0	08:29	18:38
1	1	04:00	0	0	08:29	18:38
1	1	05:00	0	0	08:29	18:38
1	1	06:00	0	0	08:29	18:38
1	1	07:00	0	0	08:29	18:38
1	1	08:00	0	0	08:29	18:38
1	1	09:00	80	59,42	08:29	18:38
1	1	10:00	75,08	50,27	08:29	18:38
1	1	11:00	66,08	39,17	08:29	18:38
1	1	12:00	59,21	25,68	08:29	18:38
1	1	13:00	55,28	9,93	08:29	18:38
1	1	14:00	54,94	-6,91	08:29	18:38
1	1	15:00	58,24	-22,98	08:29	18:38
1	1	16:00	64,64	-36,92	08:29	18:38
1	1	17:00	73,31	-48,44	08:29	18:38
1	1	18:00	80	-57,91	08:29	18:38
1	1	19:00	0	0	08:29	18:38
1	1	20:00	0	0	08:29	18:38

Figure 4.2.3: Cleaned Dataset Post-Preprocessing.

4.2.2 Model selection

a) Selecting the Optimal Model from the scikit-learn Library

Once our dataset is prepared, the next critical step is selecting the appropriate machine learning model to predict the orientation angles of the solar panel. At this stage, it is crucial to have a deep understanding of the problem and to consider all relevant details in order to identify the model that best fits the specific challenge we are addressing.
In order to choose our machine learning model, we have used the Scikit-learn library, commonly known as sklearn, which is one of the most popular and widely used machine learning libraries in Python. It provides a comprehensive set of tools, including pre-processing utilities, model training algorithms, and model evaluation methods, making it a versatile choice for various machine learning tasks.
The step that comes after is selecting the appropriate estimator, for that Scikit-learn offers a flowchart that guides users in identifying the most suitable estimator based on the nature of their problem. This flowchart helps narrow down the options by considering factors

such as the size of data, the task (classification, regression, etc.), and the specific requirements of the model.

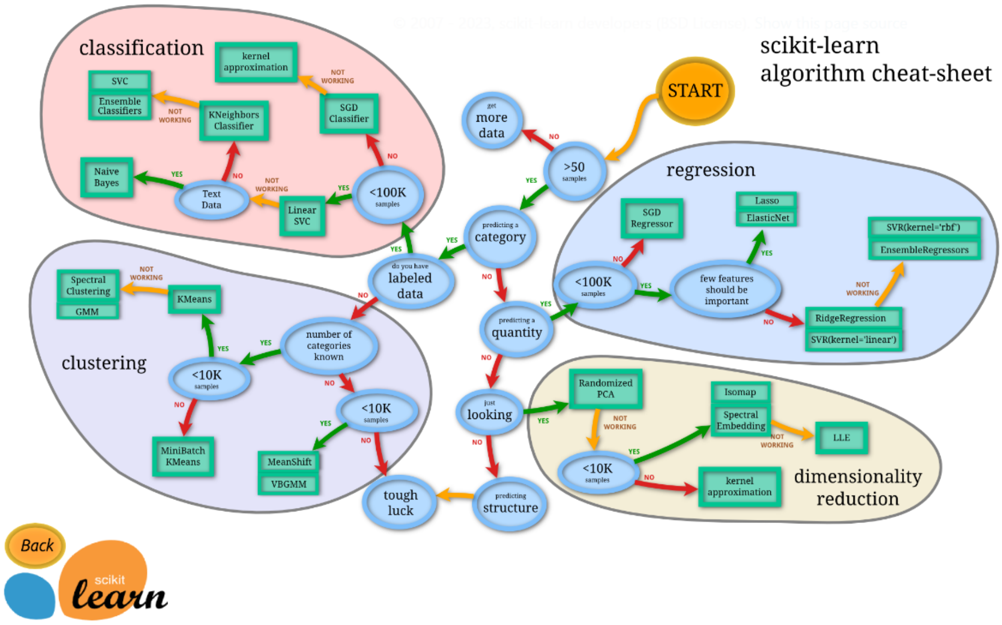


Figure 4.2.4: Scikit-learn estimators guide.

b)Ensemble learning and the wisdom of the crowd

Ensemble learning refers to a machine learning approach where several models are trained to address a common problem, and their predictions are combined to enhance the overall performance. The idea behind ensemble learning is that by combining multiple models, each with its strengths and weaknesses, the ensemble can achieve better results than any single model alone. Ensemble learning can be applied to various machine learning tasks, including classification, regression, and clustering.

This idea comes from a popular concept known as the "wisdom of the crowd", and it refers to the idea that large groups of people are collectively smarter than individual experts when it comes to problem-solving, decision-making, innovating, and predicting. The idea is that the viewpoint of an individual can inherently be biased, whereas taking the average knowledge of a crowd can result in eliminating the bias or noise to produce a clearer and more coherent result.

However, for collective performance to yield better results, and for the crowd to truly be considered wiser than the individual, it must meet certain conditions:

- The group should have dispersed knowledge or expertise. No single person should dominate the decision-making process; rather, the collective input should be drawn from various areas of expertise.
- Individuals within the group should have different perspectives and viewpoints. A homogeneous group is more likely to share the same biases, limiting the range of possible solutions.
- The group needs to be large enough to dilute individual errors or biases. A small crowd may not reflect a wide enough range of knowledge or viewpoints.

In machine learning, we can apply this concept by building ensembles of models that perform better than the best individual machine learning models.

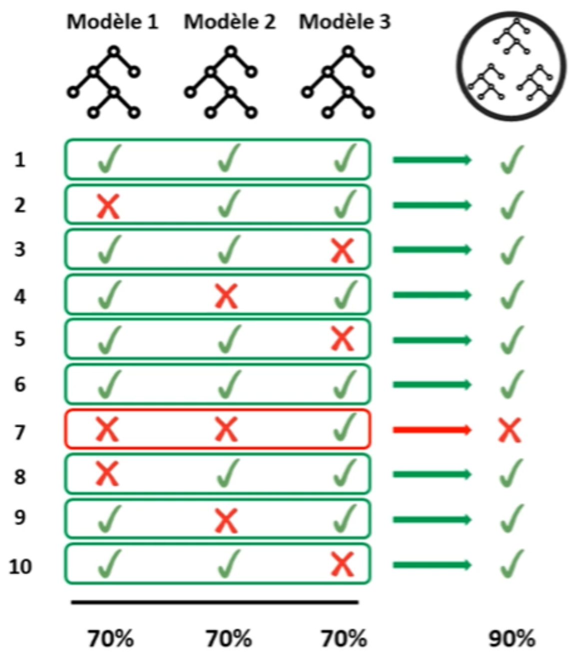


Figure 4.2.5: Optimizing performance using Ensemble Learning.

c) Ensemble learning algorithms: Bagging, Boosting and Stacking

What is bagging ?
This ensemble learning algorithm involves the following steps :

- **Data Sampling** : Creating multiple subsets of the training dataset using bootstrap sampling (random sampling with replacement).
- **Model Training** : Training a separate model on each subset of the data.
- **Aggregation** : Combining the predictions from all individual models (averaged for regression or majority voting for classification) to produce the final output.

Bagging is mainly applied to tree-based machine learning models such as decision trees and random forests.

What is Boosting ?

In boosting, we train a sequence of models. Each model is trained on a weighted training set. We assign weights based on the errors of the previous models in the sequence.
The main idea behind sequential training is to have each model correct the errors of its predecessor. This continues until the predefined number of trained models or some other criteria are met.
Boosting generally improves the accuracy of a machine learning model by improving the performance of weak learners. We typically use XGBoost, CatBoost, and AdaBoost.

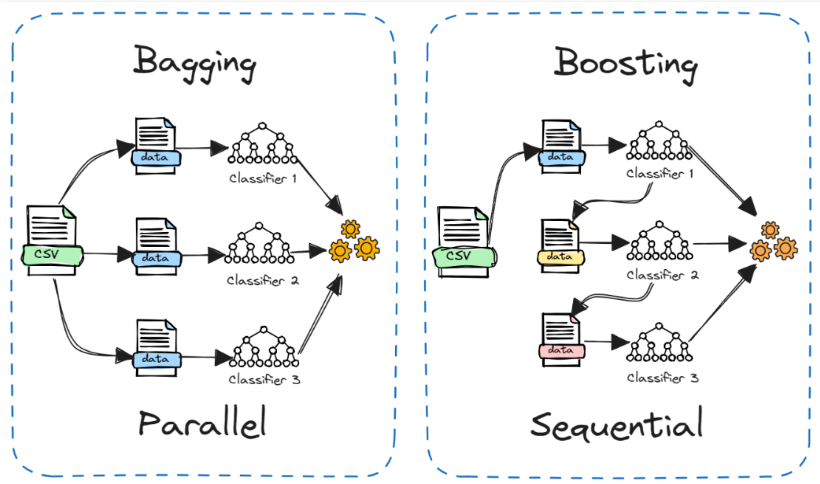


Figure 4.2.6: comparison between Bagging and Boosting algorithms.

What is stacking?

This ensemble technique works by applying input of combined multiple weak learners' predictions and Meta learners so that a better output prediction model can be achieved.
In stacking, an algorithm takes the outputs of sub-models as input and attempts to learn how to best combine the input predictions to make a better output prediction. The basic architecture of stacking can be represented as shown below the image.

Figure 4.2.7: The architecture of stacking algorithms.

d) Decision trees and random forest algorithm

As mentioned earlier, our goal is to predict the orientation angles of the solar panel using supervised learning techniques. Since the target values we aim to predict (the orientation angles) are continuous, this makes it a regression problem rather than a classification

problem.
After reviewing different machine learning models, we decided to use the Random Forest Regressor, which is part of the scikit-learn library. This model is well-suited for regression tasks due to its ability to handle complex datasets, reduce overfitting through ensemble learning, and provide accurate predictions by averaging the results of multiple decision trees. However, to understand how this estimator works, it is essential to first understand how decision tree-based algorithms operate.

1. **Decision trees :**
   In general, a decision tree is a tool used to make choices by breaking down a decision into a series of simpler, step-by-step options. Each point where a decision is made represents a branch, and each branch leads to further options or outcomes. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
   In machine learning, decision trees are a method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

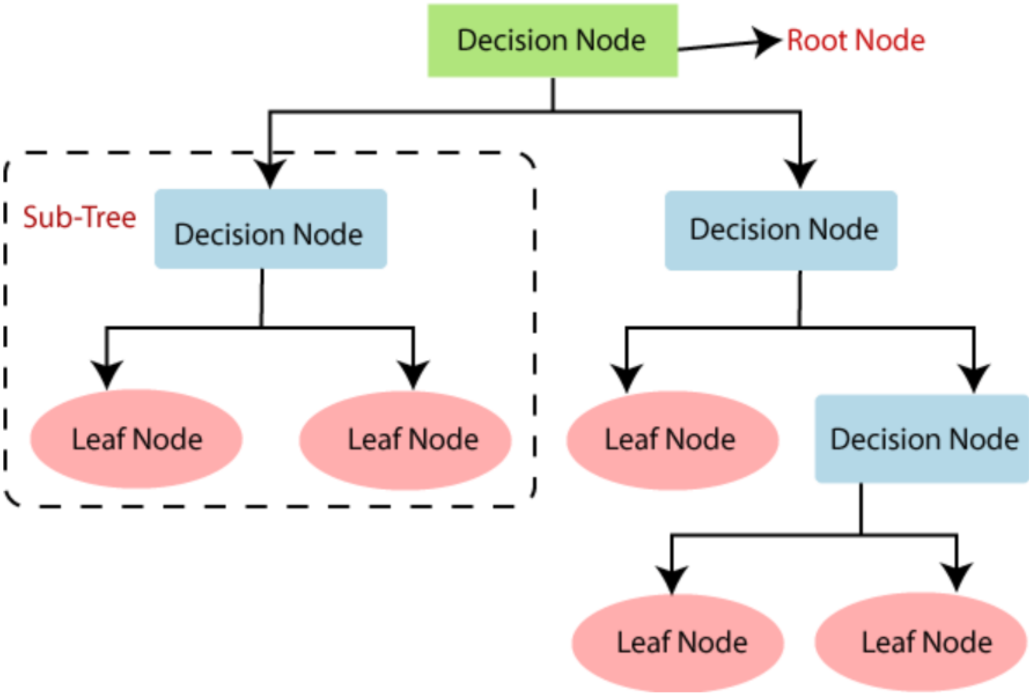


Figure 4.2.8: Fundamental elements of a decision tree.

Decision trees have many advantages:

- Simple to understand and to interpret.
- Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed.
- The cost of using the tree is logarithmic in the number of data points used to train the tree.
- Able to handle both numerical and categorical data.
- Able to handle multi-output problems.
- Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
- Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.

However, in decision tree algorithms, one of the most difficult things to manage is **overfitting**.

Overfitting occurs when the tree becomes too complex, capturing noise and small details in the training data that do not generalize well to new, unseen data. This happens when the tree grows too deep, with many branches and splits, leading to **a model that performs well on the training data but poorly on test data** .

2. **Random Forest Regressor :**
   A random forest is an ensemble learning method that combines the predictions from multiple decision trees to produce a more accurate and stable prediction.
   Random Forest Regression is an ensemble learning technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging (see paragraph above about bagging). The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.
   The Random Forest Regressor offers multiple parameters, each playing a crucial role in shaping the outcomes. Adjusting these parameters can significantly affect the model's accuracy and performance :

   n_estimators: The number of trees in the forest.
   max_depth: The maximum depth of the tree.
   min_samples_split:The minimum number of samples required to split an internal node
   min_samples_leaf:The minimum number of samples required to be at a leaf node
   max_features: The number of features to consider when looking for the best split...

4.2.3 Model training and evaluation

In machine learning, it is crucial to avoid evaluating a model's performance using the training data, as this can lead to misleading results. Instead, it is always more meaningful to assess the model on data it has never encountered, giving a better indication of how well it will perform on new, unseen examples in the future. To achieve this, we typically split the dataset into two parts: a training set containing 80% of the data and a test set with the remaining 20%. In our project, we used the **train_test_split** function from the **model_selection** module. To train our model, we use the **fit** function that receives as parameters our training set. For model evaluation, we can use the score function and other metrics such as **mean_absolute_error** and **mean_squared_error** .

4.3 Optimizing panel orientation using pvlib library

4.3.1 Introduction

Using a machine learning model to predict optimal angles for panels orientation is one option, but it's not the only way. the pvlib Python library offers a more straightforward solution. With just the geographic location and time, pvlib can directly calculate the optimal orientation to capture the most solar energy. This method uses well-established solar position algorithms, eliminating the need for complex data or model training. It provides a simple yet effective way to achieve accurate results, making it an excellent choice for solar projects that prioritize ease and precision.

4.3.2 What is pvlib ?

pvlib python is a community developed toolbox that provides a set of functions and classes for simulating the performance of photovoltaic energy systems and accomplishing related tasks. The core mission of pvlib python is to provide open, reliable, interoperable, and benchmark implementations of PV system models. The pvlib Python library contains various modules that offer a range of functions for modeling and simulating solar energy systems :

Figure 4.3.9: pvlib logo.

1. **Location and Solar Position :**

    - pvlib.location : Defines a specific location based on latitude, longitude, and altitude. It is used to calculate solar position and time.
    - pvlib.solarposition: Computes the position of the sun (solar elevation, azimuth angles) based on the time and location.

2. Clear Sky Models :

- pvlib.clearsky: Provides models to calculate clear-sky irradiance, including simplified and detailed models like the Ineichen or Hauser model.

3. Irradiance :

- pvlib.irradiance: Models the irradiance received on a tilted surface. This includes functions to calculate diffuse and direct components, plane-of-array irradiance, and tools for transforming global horizontal irradiance (GHI) to other components.

4. Atmospheric Models :

- pvlib.atmosphere: Contains functions to calculate air mass, extraterrestrial irradiance, and other atmospheric properties like linke turbidity.

5. **PV System Modeling :**

    - pvlib.pvsystem: Models the behavior of a PV system, including module performance, temperature effects, inverters, and tracking systems. It allows for detailed simulations of photovoltaic systems.

6. **Temperature Models :**

    - pvlib.temperature: Includes methods to estimate cell and module temperature, which are important for calculating performance.

7. **PV Module and Inverter Databases :**

    - pvlib.pvsystem.retrieve_sam : Provides access to databases for photovoltaic module and inverter characteristics from the System Advisor Model (SAM) repository.

8. **Tracking Systems :**

    - pvlib.tracking : Models solar tracking systems that follow the sun's movement to maximize energy output. It includes single-axis and dual-axis tracking.

4.3.3 Pvlib usage in optimizing panel orientation

As mentioned earlier, the purpose of using pvlib is to obtain the sun's position in the geographic area where our UGV operates. This is made possible by the **get_solarposition** function from the **solarposition** module. The **get_solarposition** function from the **pvlib.solarposition** module is used to calculate the position of the sun in the sky at specific times and locations. It provides key information like the solar elevation and azimuth angles, which are essential for determining the optimal orientation of solar panels.

This is the function signature :

```
pvlib.solarposition.get_solarposition(time, latitude, longitude,
altitude=None, pressure=None, method='nrel_numpy',
temperature=12,**kwargs)
```

As shown, the function has multiple parameters :

- **time** (pandas.DatetimeIndex) : Must be localized or UTC will be assumed.
- **latitude** (float) : Latitude in decimal degrees. Positive north of equator, negative to south.
- **longitude** (float) :Longitude in decimal degrees. Positive east of prime meridian, negative to west.
- **altitude** (float, optional) : If not specified, computed from pressure. Assumed to be 0 m if pressure is not supplied.
- **pressure** (float, optional): If not specified, computed from altitude. Assumed to be 101325 Pa if altitude is not supplied.
- **method** (string, default 'nrel_numpy') : "nrel_numpy" uses an implementation of the NREL SPA algorithm.
- **temperature** (float, default 12) : in °C.
- **kwargs** – Other keywords are passed to the solar position function specified by the method argument.

In our case, we started with a simulation of the UGV's geographic coordinates, and then we used the **get_solarposition** function to determine the orientation angles.

```python
def simulate_gps_data(lat_mean=31.6802337, lon_mean=-8.0440754, lat_std=0.001, lon_std=0.001):
    latitude = np.random.normal(lat_mean, lat_std)
    longitude = np.random.normal(lon_mean, lon_std)
    return latitude, longitude
```

Figure 4.3.10: simulation of the UGV's geographic coordinates.

```python
def simulate_sun_sensor(latitudes, longitudes, timestamps):
    solar_positions = []
    for lat, lon, timestamp in zip(latitudes, longitudes, timestamps):
        solar_position = pvlib.solarposition.get_solarposition(timestamp, lat, lon)
        solar_positions.append(solar_position)
    return solar_positions
```

Figure 4.3.11: Using the get_solarposition function to determine the panel orientation.

4.4 Interpretation and results exploitation

4.4.1 Light Sensor Integration for Efficient Panel Orientation

Our UGV is equipped with light sensors, which are intended **to optimize the performance of the solar panel**. Instead of relying on a basic sun-tracking system that simply follows the sun's position throughout the day, we propose a more advanced approach. This system will intelligently manage the panel's orientation by integrating data from the light sensors. Specifically, it will be able to detect when weather conditions are unfavorable, such as during rain or overcast periods. In such cases, the system will automatically halt the panel's movement, preventing unnecessary energy consumption by the actuators responsible for adjusting the panel. This ensures that the solar panel only operates when there is sufficient solar radiation for effective energy production, thereby optimizing energy use and prolonging the lifespan of the actuators.
To test this approach, we set a minimum **light level of 200 lux** as the threshold for sufficient solar radiation. This means that the system will only move the solar panel or make adjustments when the light intensity reaches at least 200 lux. This level indicates that there is enough sunlight to generate energy effectively.

```python
def simulate_actuator_movements(solar_positions, current_light_level):
    panel_angles = []
    for _, position in solar_positions.iterrows():
        if position['elevation'] < 0 or current_light_level < 200:
            elevation_angle = 0
            azimuth_angle = 0
        else:
            elevation_angle = position['elevation']
            azimuth_angle = position['azimuth']
        panel_angles.append((elevation_angle, azimuth_angle))
    return panel_angles
```

Figure 4.4.12: function for improving energy efficiency using light sensors.

By using this simple approach, we ensure that the solar panel is only operated when it can actually produce usable energy, which helps in making the system more efficient and reducing unnecessary energy use.

4.4.2 Adaptation of the solar panel orientation to the vehicle's movement

Unlike most sun-tracking systems that are fixed in place, ours is mounted on a moving vehicle. This adds an extra layer of complexity when trying to keep the solar panel properly aligned with the sun. To address this challenge, we came up with a simple solution: adjust the panel's orientation to cancel out the effect of the vehicle's rotation.

The idea is straightforward: when the vehicle turns, the solar panel rotates by the same angle in the opposite direction. For example, if the panel is set at an angle of x and the UGV turns by an angle of y, the new orientation of the panel will be adjusted to (x-y).

This way, the movement of the vehicle doesn't interfere with the solar panel's position, allowing it to stay aligned with the sun and maximize energy capture.

- **determination of the rotation angle of the UGV**
  To be able to detect the rotation angles of the UGV, it must be equipped with a gyroscope. This sensor measures the angular velocity of the vehicle, allowing us to track its movements and determine how much it has rotated over time. By integrating this data, we can accurately calculate the current rotation

angle, which is essential for adjusting the solar panel's orientation and ensuring it stays aligned with the sun, regardless of the vehicle's movement.

Figure 4.4.13: image of classic gyroscope.

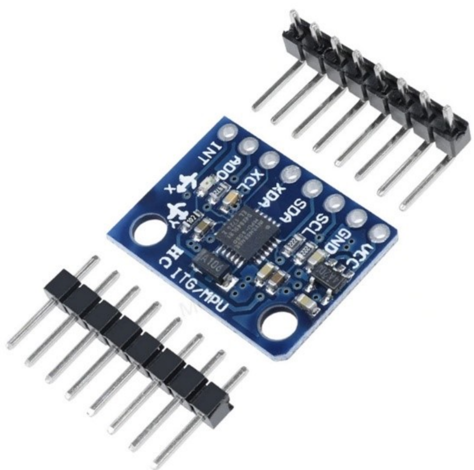


Figure 4.4.14: MPU6050 gyroscope and accelerometer.

In our code, we have started with a simulation of the data received from the gyroscope :

```python
def simulate_imu_data(accel_mean=0, accel_std=1, gyro_mean=0, gyro_std=0.1, mag_mean=0, mag_std=1):
    accel_data = np.random.normal(accel_mean, accel_std, 3)
    gyro_data = np.random.normal(gyro_mean, gyro_std, 3)
    mag_data = np.random.normal(mag_mean, mag_std, 3)
    return accel_data, gyro_data, mag_data

def calculate_orientation(accel_data, mag_data):
    ax, ay, az = accel_data
    mx, my, mz = mag_data
    roll = np.arctan2(ay, az)
    pitch = np.arctan2(-ax, np.sqrt(ay**2 + az**2))
    mag_x = mx * np.cos(pitch) + mz * np.sin(pitch)
    mag_y = mx * np.sin(roll) * np.sin(pitch) + my * np.cos(roll) - mz * np.sin(roll) * np.cos(pitch)
    yaw = np.arctan2(-mag_y, mag_x)
    return np.degrees(roll), np.degrees(pitch), np.degrees(yaw)
```

Figure 4.4.15: simulation code.

Afterward, we utilized this data to determine the orientation angles of the solar panel.

```python
def process_angles(elevation_angle, azimuth_angle, yaw, pitch):
    if elevation_angle != 0 and azimuth_angle != 0:
        compensated_elevation = elevation_angle - pitch
        compensated_azimuth = azimuth_angle - yaw
    else:
        compensated_elevation = 0
        compensated_azimuth = 0
    return compensated_elevation, compensated_azimuth
```

Figure 4.4.16: simulation code.

4.5 Conclusion

Optimizing the solar energy produced through adjusting the solar panel's orientation is not a simple goal to achieve, but there are numerous approaches to reach it. In this project, we presented two major solutions: one based on using a machine learning model and the other utilizing the pvlib library in Python. Both solutions yielded convincing results, but pvlib is more advantageous as it does not require any specific prerequisites (such as dataset preparation or model selection) and provides direct and accurate results. Finding the best panel orientation was not the only goal of this project. The examination of various methods to optimize our system was even more important. Leveraging simple data, such as information from light sensors or the gyroscope, could add significant value to our work, reflecting our desire to learn and explore every possible means to improve our work.

Chapter 5

Real-Time Monitoring with Interactive Dashboards

5.1 Introduction

Real-time monitoring of Unmanned Ground Vehicles (UGVs) has reached a new level of sophistication with the development of an interactive dashboard using Streamlit. This dashboard marks a significant advancement by combining cutting-edge technology with practical application, transforming how UGV data is visualized and managed.
In this chapter, we explore the intricacies of this innovative dashboard, starting with its design and the various sections that enhance user interaction and data insights. We delve into the API infrastructure that ensures efficient data management and seamless integration between the UGVs and the dashboard. This infrastructure not only supports real-time data collection and processing but also facilitates dynamic visualization.
By leveraging technologies like Streamlit and advanced API frameworks, this chapter illustrates the transformative potential of real-time monitoring systems in enhancing operational efficiency and decision-making for UGVs.

5.2 Designing the User Interface with Streamlit

The user interface was designed using Streamlit, a tool that allows the creation of interactive and user-friendly web applications. The dashboard is divided into several main sections: solar panel monitoring, UGV monitoring, environmental data monitoring, and a section dedicated to reports and alerts. The main code file for this interface is **dashboard_main.py**, which manages the display of all sections from a simple navigation menu. This file serves as the entry point for the user and centralizes all the dashboard functionalities.

5.2.1 Objectives of the Interface

- Accessibility and intuitiveness : The interface should be simple to navigate, even for non-technical users.

- Real-time display: Data should be continuously updated, ensuring optimal responsiveness.

- Modularity: The separation of different sections (solar panels, UGV, environment) for clear and distinct data management.

Interface Sections :

- Solar Panel Monitoring
- UGV Monitoring
- Environmental Data
- Reports and Alerts

Overview of the User Interface

This figure shows the main interface with a navigation menu allowing access to the different monitoring sections. See figure 5.2.1.

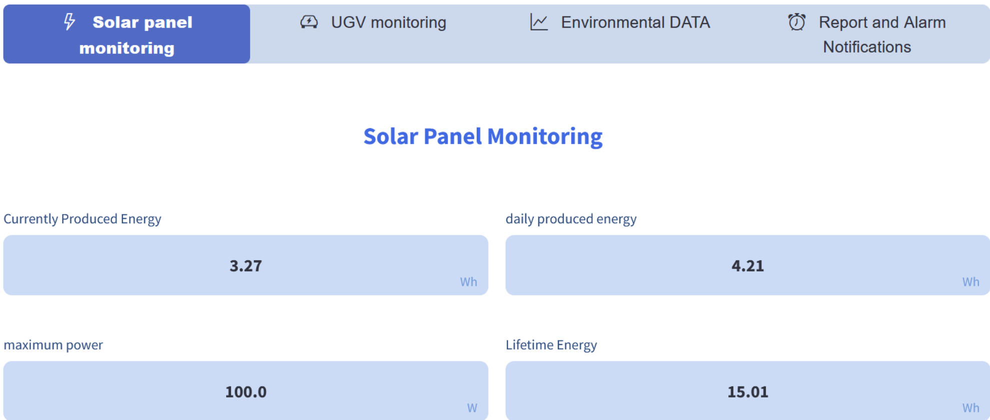


Figure 5.2.1: Dashboard showing the different sections.

5.2.2 Code File

dashboard_main.py is responsible for setting up the user interface and managing the flow between the different dashboard sections. It includes a navigation menu that allows switching between the solar panel monitoring, UGV monitoring, environmental data, and alerts modules.

• dashboard_main.py interacts with other code files responsible for data collection and their visualization in each section.

5.2.3 Developing the API Infrastructure

The API infrastructure is developed using FastAPI to enable communication between the UGV and the dashboard. The API handles the real-time sending and receiving of sensor data. It allows the dashboard to access sensor data for real-time display.

What is API ?

APIs–Application Programming Interfaces–are the building blocks of modern technology and business infrastructure. At its most basic level, an API is a piece of software that allows two applications to talk to each other. APIs connect apps to apps, apps to servers, and apps and servers to consumers.

What does API stand for ?

API stands for Application Programming Interface. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses. Their API documentation contains information on how developers are to structure those requests and responses.

How do APIs work ?

API architecture is usually explained in terms of client and server. The application sending the request is called the client, and the application sending the response is called the server. So in the weather example, the bureau's weather database is the server, and the mobile app is the client.

What is FastAPI ?

FastAPI is a state-of-the-art Python web system made to make it more straightforward to make superior execution APIs.Because of its speed, type hinting support, automatic documentation creation, and asynchronous features, it was developed by Sebastián Ramrez and has grown in popularity.

Characteristics of FastAPI

- Automatic Documentation : Based on the type hints and docstrings in your code, FastAPI automatically creates interactive documentation for your API using programs.
- Type Hinting: Built-in Python type hints support guarantees reliable auto-completion and validation in contemporary IDEs, enhancing code dependability.
- Fast Performance: One of the quickest Python web frameworks available, FastAPI is built for speed. It takes advantage of nonconcurrent programming to successfully deal with different solicitations immediately.
- Easy Validation: Request and input data are automatically verified, which reduces the likelihood of errors. A selection of validation techniques is offered by FastAPI to guarantee data consistency.
- Reliance infusion empowers measured and viable programming by simplifying it to infuse conditions into Programming interface endpoints.
- Authentication and Authorization: FastAPI supports OAuth2, JWT, and other protocols, making it simple to implement authentication and authorization processes.

Code files managing the API

- **api_server.py** : This file is the API server that receives the data sent by the UGV and temporarily stores them for retrieval by the dashboard.
- **sensor_data.py** : This file defines the data structures and types of information sent by the UGV (e.g., temperature, humidity, etc.).

Relationships

- **api_server.py** receives data from the UGV and stores it, while dashboard_main.py retrieves this data from the API to display it in the corresponding sections of the dashboard.

API Infrastructure Diagram

This diagram shows how the UGV sends data to the API and how it is displayed in real-time on the dashboard. See figure 5.2.2.

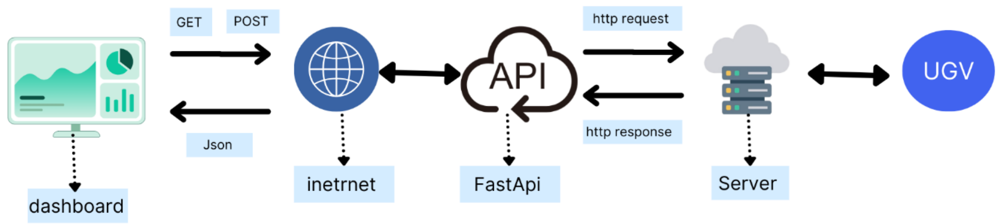


Figure 5.2.2: Dashboard showing the different sections.

5.2.4 Solar Panel Monitoring

The solar panel monitoring module presents information such as the energy produced by the solar panels, their orientation, and the energy production history.
The file associated with this section is solar_panel_monitoring.py. It is responsible for collecting solar sensor data and generating visualizations to display the energy produced, the panel angle, and historical production graphs.

Relationship

solar_panel_monitoring.py retrieves data from the API via dashboard_main.py and displays it on the dashboard.

Solar Panel Monitoring

The figure shows real-time graphs and metrics related to the solar panels' energy production.

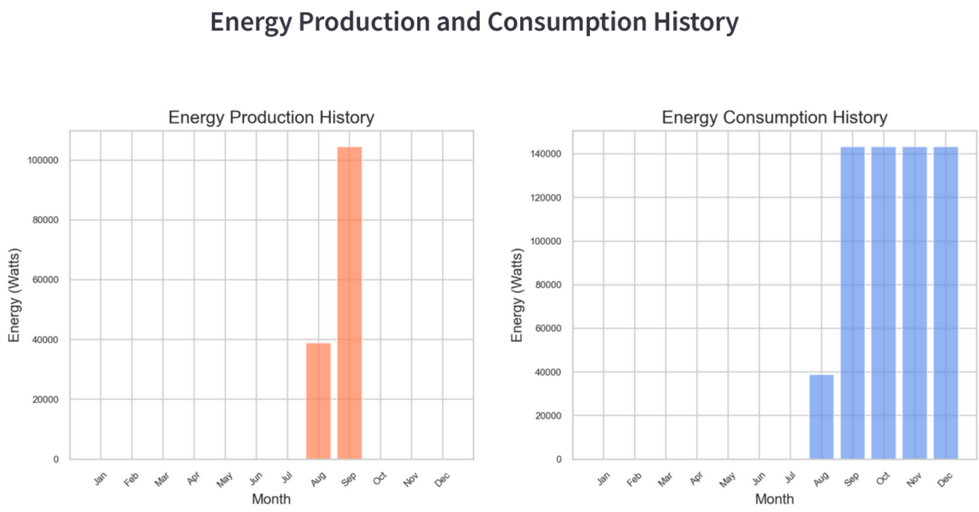


Figure 5.2.3: Real-time graphs and metrics related to solar energy production.

5.2.5 UGV Monitoring

The UGV monitoring module allows real-time tracking of the vehicle's performance, including speed, battery level, and GPS position. An interactive map is also displayed to track the vehicle's location. The file ugv_monitoring.py handles the collection of data such as speed and GPS position and their visualization on the dashboard. This file also interacts with libraries to display a dynamic map.

Relationships

UGV_monitoring.py retrieves data sent by the UGV via the API and displays it on the dashboard in collaboration with **dashboard_main.py**.

UGV Monitoring

This figure shows the UGV's speed, GPS position data, battery level, and operating time, along with a map displaying the real-time location.

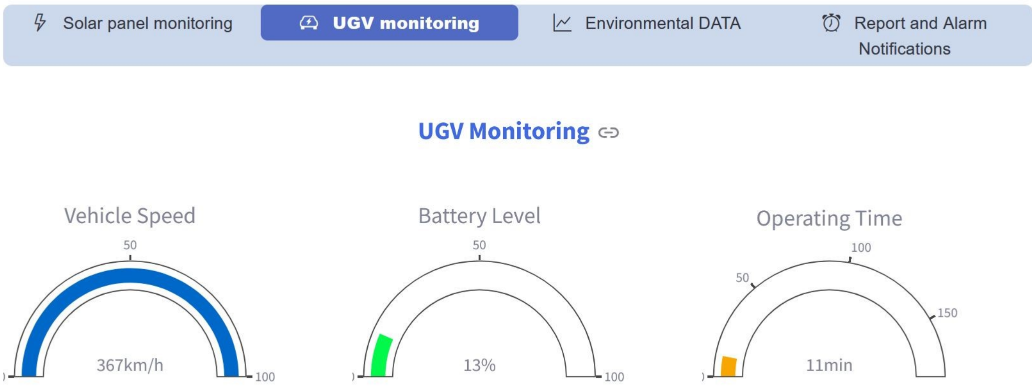


Figure 5.2.4: UGV Monitoring part.

5.2.6 Environmental Data

This module is dedicated to visualizing environmental data, such as temperature, humidity, and light levels, captured in real-time. These data are essential to understanding the impact of weather conditions on the UGV's performance.
The file environmental_data_monitoring.py is responsible for collecting environmental data and displaying it as graphs and metrics on the dashboard.

Relationships

environmental_data_monitoring.py retrieves data via the API and generates graphs for visualization on the dashboard.

Environmental Data Monitoring

The figure shows temperature, humidity, and light data, along with graphs illustrating the evolution of these values.

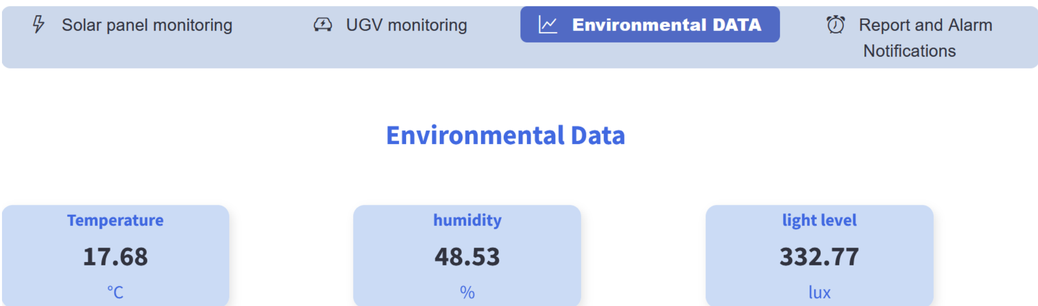


Figure 5.2.5: Real-time graphs and metrics related to solar energy production.

5.2.7 Reports and Alerts

This module is responsible for generating automatic reports based on the collected data and sending real-time alerts when a critical condition, such as a low battery level, is detected. The file reports_alerts.py handles the logic for sending alerts and generating weekly or daily reports based on the data collected by the other modules.

Relationships

reports_alerts.py interacts with data from solar_panel_monitoring.py, ugv_monitoring.py, and environmental_data_monitoring.py to generate comprehensive reports and send alerts via dashboard_main.py.

Reports and Alerts

This figure shows a low battery alert and an example of a report automatically generated from the collected data.

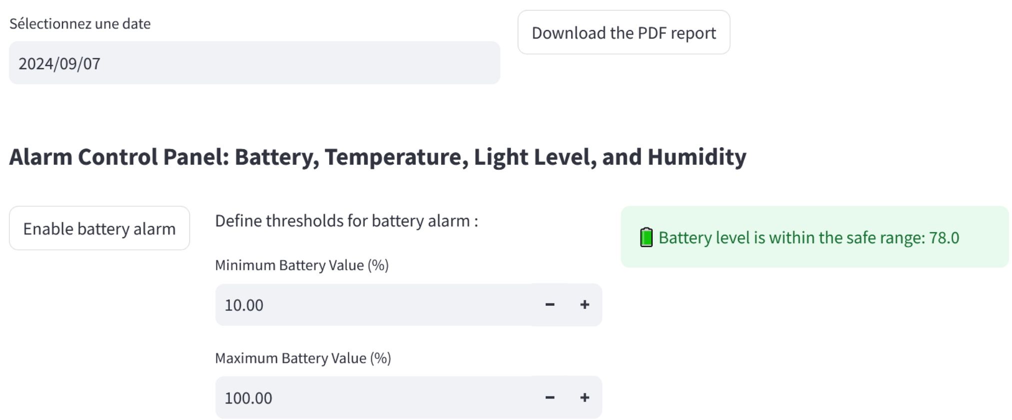


Figure 5.2.6: Battery alert.

5.2.8 Overview of the Dashboard

The interactive dashboard is designed to centralize all the necessary information for monitoring the UGV. Each module is interconnected, and users can easily navigate between the different sections via a menu.

Relationships between the files

- **dashboard_main.py** : The main file that controls the user interface.
- **api_server.py** : Manages data from the UGV and makes it available to the dashboard.
- **solar_panel_monitoring.py, ugv_monitoring.py, environmental_data_monitoring.py** and **reports_alerts.py** : Retrieve data from the API and display the corresponding information in their respective sections.

5.3 Conclusion

This chapter explains how real-time monitoring of the UGV (Unmanned Ground Vehicle) was implemented using an interactive dashboard developed with Streamlit. It details the various sections of the dashboard, the API infrastructure used to manage the data, and the relationship between the different code files.

General conclusion

This project has successfully addressed the challenge of optimizing solar energy harvesting for Unmanned Ground Vehicles (UGVs) equipped with solar panels and lithium batteries. By developing and implementing an advanced system designed to maximize solar energy collection, we have significantly extended the UGV's operational duration and enhanced its performance and sustainability. The project involved comprehensive foundational research, an in-depth exploration of solar energy technologies, and the integration of machine learning techniques to optimize energy management. Additionally, the real-time monitoring system has provided ongoing insights and adjustments to ensure the system operates efficiently under varying conditions.

Our group of seven has greatly benefited from this collaborative project. Working together on this complex challenge has offered us invaluable hands-on experience with cutting-edge technologies. We have gained a deeper understanding of solar energy systems, machine learning, and their practical applications in autonomous vehicles. This collective effort has not only strengthened our technical skills but also improved our ability to collaborate effectively as a team. The experience has been instrumental in preparing us for future professional challenges, equipping us with both advanced technical knowledge and enhanced teamwork capabilities.

Bibliography

https://www.simplilearn.com/what-is-fastapi-article

https://aws.amazon.com/what-is/api/?nc1=h_ls

https://cloud.google.com/learn/what-is-api-management?hl=en

https://freyrenergy.com/solar-panel-efficiency-dont-let-your-panels-

https://www.8051projects.net/files/public/1273145323_27693_FT32042_solatauto.pdf

https://www.onepointesolutions.com/blog/solar-energy-applications/

https://www.britannica.com/science/solar-energy

https://aws.amazon.com/what-is/scrum/

https://www.investopedia.com/terms/w/wisdom-crowds.asp

https://www.investopedia.com/terms/w/wisdom-crowds.asp

https://scikit-learn.org/1.3/tutorial/machine_learning_map/index.html

https://www.baeldung.com/cs/bagging-boosting-stacking-ml-ensemble-mo

https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-metho

https://pvlib-python.readthedocs.io/en/stable/

https://www.hlevkin.com/hlevkin/45MachineDeepLearning/ML/
Foundations_of_Machine_Learning.pdf

https://pdfcoffee.com/download/machine-learning-zhi-hua-zhou-3-pdf-f
html