Development of an automated space rated solar panel array for future lunar rover mission

CECILIA MARTÍNEZ MARTÍN
Development of an automated space rated solar panel array for future lunar rover mission

CECILIA MARTÍNEZ MARTÍN
Abstract

Space industry has been enormously growing in the last decades. Energy harvesting for space applications is usually a critical aspect of space missions, and therefore a method for doing it is included in most of the satellites and vehicles sent to space. This project focuses on energy harvesting for a Moon Rover. This will be done via a deployable solar panel system, which will be orientable in order to maximize the efficiency.

This project is divided in two parts. The first one will be the mechanical design of the deployment system. The main deployment will be accomplished via a composite material boom, which will pull up a solar panel array. A second deployment phase will be accomplished using flaps that will slide out of the main system.

The second part of the project will focus on the control of the panels. This task will be achieved using Model Predictive Control (MPC). This control will find the optimal way of moving the panels so that the energy is efficiently used.

Sammanfattning


Den andra delen av projektet kommer att fokusera på kontrollen av panelerna. Det här uppgiften uppnås med hjälp av Model Predictive Control (MPC). Den här kontrollen kommer hitta det optimala sättet att flytta panelerna så att energin används effektivt.
Acknowledgements

I would like to thank the people at Space Applications Services for giving me the great opportunity of participating in one of their projects and for trusting me with the developed solution. Specially Daniel Fodorcan, my supervisor, who has helped me whenever he could, and has given me a lot of useful advice on the hardware part. I would also like to thank Alin Munteanu, who also helped me with the hardware development, and Mathieu Deremetz, who gave me support for both parts of the project. In general, everybody has been very friendly and welcoming, and I appreciate very much all the support received during the project.

I would also like to thank Dimos Dimarogonas, my examiner, who has given me very useful guidelines and suggestions on the report in order to have a nice project in the end. Thanks also to Lars Lindemann and Pedro Roque, my two supervisors. They have been very good help, despite the long distance. They have given me very useful tips on how to continue so that I could have the report on time.

I would also like to thank KTH and ETSII, UPM for giving me the opportunity of carrying out the double degree program in which I am taking part. It has been a very interesting experience, which I would not have wanted to miss. I also want to thank my friends, the ones that I had before starting this adventure, and the new ones that I met during my stay in Stockholm. They have made the journey easier and funnier.

Last but not least, I would like to thank my parents, Andrés Martínez and Raquel Martín, for all the support that they have given me during my stay abroad, and for believing in me during my whole life. I would not be here today if it weren’t for them, and I want to thank them for that. I would also like to thank my sisters Ana and Laura Martínez Martín for supporting me in everything.
Contents

1 Introduction 3
   1.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
   1.2 Thesis outline . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

I Mechanical Design 5

2 Introduction 6
   2.1 Background and aim . . . . . . . . . . . . . . . . . . . . . . . . . 6
   2.2 State of the art in deployable systems . . . . . . . . . . . . . . . 7
   2.3 Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3 Implementation 16
   3.1 System to implement . . . . . . . . . . . . . . . . . . . . . . . . . 16
   3.2 Solar panel storage unit . . . . . . . . . . . . . . . . . . . . . . . 17
      3.2.1 Base plate . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
      3.2.2 Shaft . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19
      3.2.3 Motor and coupling . . . . . . . . . . . . . . . . . . . . . . 19
      3.2.4 Solar cells' support . . . . . . . . . . . . . . . . . . . . . . 22
      3.2.5 Complete base system . . . . . . . . . . . . . . . . . . . . . 23
   3.3 Solar panel lateral deployment . . . . . . . . . . . . . . . . . . . . 23
      3.3.1 Base idea . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
      3.3.2 Design of the system . . . . . . . . . . . . . . . . . . . . . . 25
   3.4 Boom system . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
      3.4.1 Design . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
      3.4.2 Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27
   3.5 Complete system . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
      3.5.1 Coupling between the parts . . . . . . . . . . . . . . . . . . . 28

4 Results and Conclusions 29

II Control 31

5 Introduction 32
   5.1 Background . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
   5.2 Aim . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
   5.3 State of the art in solar tracking systems . . . . . . . . . . . . . . 33
   5.4 Requirements . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
## CONTENTS

6 Implementation ................................................. 40
6.1 Algorithm to implement .................................. 40
6.2 Base idea .................................................. 40
6.3 Models ...................................................... 42
   6.3.1 Ideal Panel Trajectory ................................. 42
   6.3.2 Generated Energy ...................................... 45
   6.3.3 Consumed Energy ...................................... 47
   6.3.4 Motor model ........................................... 47
6.4 Modifications to the models .............................. 50
   6.4.1 Generated power ....................................... 50
   6.4.2 Motor model ........................................... 51
6.5 MPC formulation ........................................... 52

7 Experiments ..................................................... 53
7.1 Tools ....................................................... 53
7.2 Scenarios and simulations ................................. 56
   7.2.1 Initial simulations ..................................... 56
   7.2.2 Angle management ..................................... 60
   7.2.3 Stop moving flag implementation ....................... 62
   7.2.4 Current initialization .................................. 67
   7.2.5 Disturbance ............................................ 68
   7.2.6 Complete simulation .................................... 69

8 Results and Conclusions .................................... 72

III Results and Conclusions .................................. 76

9 Results ......................................................... 77

10 Conclusions .................................................. 80
   10.1 Future work .............................................. 81

Bibliography ..................................................... 83

List of Figures .................................................. 86

List of Tables ................................................... 89

A Listings ....................................................... 90

B Drawings ....................................................... 93
Chapter 1

Introduction

In the last years, space industry has experienced a huge growth. This is due to the great interest for space exploration that arises in the mankind. The growth of technology has allowed humans to go beyond what was possibly imaginable some decades ago.

In order to be able to explore the space, some solutions have been developed. There is a wide variety of solutions, depending on the goal of the explorations. Lately, the existence of water in the Moon has been proved, and the next question to solve is how to get it. The water found lies in places which are never hit by the sunlight. This means, in deep craters on the poles of the Moon. Therefore, companies are designing vehicles that can go to this places and find ways to extract it.

1.1 Background

This project has been carried out at Space Applications Services, a Belgian company specialized in space. The company already had an existing vehicle, but wanted to modify it in order to make it lighter and smaller. One other goal of the modifications comprised equipping it with a source of energy. The chosen source was solar photovoltaic power, but the implementation had to be done.

In order to maximize the energy obtained, the solar panels had to be movable in order to follow the Sun’s position through the whole mission.

The thesis has been developed following the requirements of the company with respect to the existing material, as well as the requirements regarding the implementation of the system.
1.2 Thesis outline

This thesis work is divided in two main parts: hardware (Part I) and software (Part II). These parts are followed by a third one which includes an overview of the results and conclusions for each of the parts and for the project as a whole.

Both parts start with an introductory chapter (Chapter 2 for the hardware and Chapter 5 for the software), in which the background and the aim of the project with regards to the part is examined. In these chapters, a state of the art is studied, and the requirements of the system are presented.

The introductory chapters are followed by a chapter about the implementation of the solution (Chapter 3 for the hardware and Chapter 6 for the software). These chapters start with the presentation of the decision of the system to be implemented, followed by its development.

Chapter 7 presents the experiments done via software to the system. This chapter is included in Part II, but uses information of Part I in what refers to the simulated system. In this chapter, some modifications to the solutions of Chapter 6 are done.

At the end of each part, there is a chapter for results and conclusions (Chapter 4 for the hardware and Chapter 8 for the software) in which the final solution is presented. The validity of the solutions is also discussed in these chapters.

To finish with the thesis, Part III is developed, with a summary of the previously obtained results, conclusions for each part and the project as a whole, and the future work proposed.

In the annex, the code used through the project, as well as some drawings of the designed parts, can be found.
Part I

Mechanical Design
Chapter 2

Introduction

To begin with the hardware part of the project, a brief introduction is made. First, the background and aim regarding hardware issues will be analyzed. The state of the art will be studied in order to find a suitable solution to the given problem. To finish with this chapter, a brief list of the requirements needed will be presented.

2.1 Background and aim

At the beginning of the project, there was already an existing rover. Nevertheless, due to time constraints, this version did not include any energy source apart from a battery. Currently, a new version of the rover is being designed. This new version includes some changes in its structure in order to make it smaller. This new version will also be equipped with solar panels so that it has a longer lifespan during the mission time.

The mission will be to the south pole of the moon, unlike the usual missions that go to the equator. This factor will imply constraints mainly in the control phase of the project, but also in the design of the system.

The aim of the project is therefore to include a solar panel deployment system that can fit inside the structure of the already existing rover. The system should be compact for the launching phase, and deploy when it arrives to the ground so that there is enough area of solar panel to provide the necessary energy to the vehicle. This way, the solar system can be compact when needed, and have bigger area for energy production.

For the design of the system, design constraints related to both the vehicle (size and shape) and the requirements (main orientation of the panels) will be taken into consideration.
2.2 State of the art in deployable systems

In order to be able to choose the appropriate deployment system for the panels, a study of the systems already used in space has been done. This study has been mainly focused on deployable systems in general, but also in deployment systems for solar panels. In the following pages, the studied systems will be presented, as well as a comparison table at the end of the section (Table 2.1). Most of the documentation found focuses on deployment systems for small satellites, but the majority of these systems can also be applied to lunar vehicles.

The aim of having a deployable structure has its basis on the idea of having a high packing ratio so that they are compact when stowed and can easily be included in the satellite or vehicle without taking much space of it, but still having the needed surface when needed. The deployment can be either active or passive. In many cases a passive system is preferred because they do not consume power, but also because their mass is usually lower, as they do not need a control system and a motor to be deployed [1].

Articulated structures

These group of structures are based on the use of hinges that can be both active or passive [2]. These structures have a high packing ratio, but also a high mass. That is the reason why this kind of structure has been mainly used in large satellites. Nevertheless, they have also been used in some small satellites. Some examples of satellites using this deployment system are XSAS (Figure 2.1), which had a stowed size of 10cm x 10cm x15 cm and reached up to two meters when deployed [3], and FalconSAT-7 (Figure 2.2).

The structure is a scissor mechanism (pantographic structure), like the one shown on Figure 2.3, which has hinges on the articulations. The hinges can be preloaded so that the system deploys by itself when it is released (as the XSAS system), or the hinges can just act as articulation, having a separate mechanism that deploys the system.

Articulated truss structures

This kind of boom gives a high strength and stiffness of the mast. Nevertheless, the mechanism usually requires a cannister for packaging, which makes it too big and heavy for small satellites. Some examples of the structures are ATK’s CoilABLE masts (Figure 2.4a), which uses carbon fiber and has a stowage length of maximum 2% of the deployment length without taking into consideration the cannister size; the ADAM-Mast (Figure 2.4b), which has rectangular section and hinges in the articulations, and was developed for applications where high accuracy is needed; and the AstroMast (Figure 2.4c), which is based on ATK’s CoilABLE mast, but twisted an angle when deployed [6].
Inflatable booms

This is a very lightweight system, and has a high packing ratio as well. The system is deployed by the introduction of a gas inside a tube that acts as skin. After the inflation, the skin is rigidized. It has some disadvantages such as low accuracy when deployed, and that the loads that it is able to deploy are low. This deployment method has been tested, for example, in the InflateSail project, shown in Figure 2.5, or by the Jet Propulsion Laboratory (JPL) in [7].

Torsion springs

They are commonly used as a passive solution where the deployment is accomplished with rotatory motion. The main advantage is that there is no
power consumption on the deployment phase, as well as their linear behavior and small volume and weight. Nevertheless, there are also disadvantages for this system, such as lack of blocking in the final phase, which leads to oscillations. This can be implemented, but the complexity, volume and weight of the system would increase.

This system has been used for the deployment of the solar panels in the QuakeSat (Figure 2.6) and a locking solution to solve the aforementioned problem has been studied in [1]. A detailed analysis of the system applied to
solar panel deployment can be found in [9].

**Telescoping boom**

This mechanism consists of several tubes with different diameters so that they can be stowed one inside the other. Therefore, when packed, the space taken by the boom is equal to the volume inside the biggest cylinder. When the system is deployed, the length of the boom is several times bigger than the length in the stowed position.

This deployment system has been used in [8] for the QuakeSat project, which also used spring loaded hinges for the deployment of the solar panels. The mechanism corresponding to the deployment of the solar panels is explained in the subsection **Torsion springs**. The QuakeSat is represented in [Figure 2.6](#).

Other example of satellite using this method is in the REXUS 9 sounding rocket. This system had three deployable sections and the deployment was caused by a pyrotechnic detonation [10]. In this case the experiment failed because the hatch mechanism did not open correctly. But the experiment done with the REXUS 11 sounding rocket was successful after the replacement of the hatch mechanism for a different one, proving the validity of this deployment system [11].

![Figure 2.6: QuakeSat in deployed configuration](image)

**Wire boom**

This deployment system has been used for several spacecraft missions. It is a solution when the goal is the deployment of weighted sensors from spinning spacecrafts [13]. For this goal, a flexible wire is used in such a way that the rotatory motion of the spacecraft creates a force that pulls outwards the sensor at the end of the wire. This system has been tested on several missions such as DICE, THEMIS, MEFISTO-S and others. The system has a rotatory damper to avoid the wire from re-wrapping. The main disadvantage of this method is that there is no control over the orientation of the sensor (or solar panel) attached to the end of the wire, along with the need of rotatory motion to deploy [14].
Origami foldable structure

This method was mainly studied for application on solar sails in the beginning, as they needed a flat surface in order to work properly and the deployment with boom mechanisms a high tension was needed in order to accomplish it. By using rotation as tension to keep the surface flat and a folded structure no boom was needed anymore and the system was simpler and lighter [14]. The system works using a skew fold as shown in Figure 2.7.

![Figure 2.7: Skew fold configurations][14]

This method can also be used to deploy solar panels. Nevertheless, the thickness of the panels has to be taken into consideration in order to be able to stow the system. This study was done in [15] and an equivalent wooden model with the correct gaps between the plates was built (Figure 2.8).

![Figure 2.8: Wooden model of solar origami deployment system][15]

Tape spring based structures

This kind of boom can have several shapes and can also be built with different materials (either metals or composites). In all the cases, the common fact is that the way of storage is rolled in a cylinder, and when they deploy they get a curved section, so that the stiffness is high. This boom category is therefore
very light and compact in general. The main shapes that these booms have are shown in Figure 2.9 [6].

(a) STEM Booms, different configurations
(b) DLR Boom
(c) TRAC Boom

Figure 2.9: Different configurations of tape spring booms [6]

An example of the usage of this deployment method can be found in [16]. For the application, the system had to deploy synchronously in the three dimensions. To accomplish the synchronous requirement, six booms were used in a way in which the geometric constraints were enough to accomplish synchronous movement. For the deployment just the energy stowed in the boom was enough to "unroll" it and deploy. The system is shown in Figure 2.10.

Figure 2.10: Synchronous system for deployment with tape spring composite [16]
Another example of the usage of this deployment mechanism is in [17], where the boom was an antenna itself. The deployment was also caused by the energy stored in the rolled structure. The system is shown in Figure 2.11.

![Antenna deployment with composite boom][17]

Figure 2.11: Antenna deployment with composite boom [17]

In the previous cases, the energy stored in the boom was enough to deploy it. Nevertheless, when there is a need to lift a load, it is very likely that a motor to activate the deployment is needed, as the energy stored might not be enough. It is also important to consider that when the tape spring is stored for long times, there are some relaxation effects in the structure so that the energy for deployment is reduced (or might even disappear completely, depending on the time and the structure) [18] [19].

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Pros</th>
<th>Cons</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Articulated structure</td>
<td>-High packing ratio</td>
<td>-High mass</td>
<td>XSAS, FalconSAT-7</td>
</tr>
<tr>
<td></td>
<td>-Simple</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-Reliable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Articulated truss structure</td>
<td>-Great extension</td>
<td>-Mechanically complex</td>
<td>ISS, PRISM</td>
</tr>
<tr>
<td></td>
<td>-High strength and stiffness</td>
<td>-High mass</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-Reliable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inflatable boom</td>
<td>-Lightweight</td>
<td>-Need for gas tank</td>
<td>InflateSail</td>
</tr>
<tr>
<td></td>
<td>-Very compact</td>
<td>-Low accuracy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Low loads</td>
<td></td>
</tr>
<tr>
<td>Torsion spring</td>
<td>-Passive</td>
<td>-No blocking in the final phase</td>
<td>QuakeSat</td>
</tr>
<tr>
<td></td>
<td>-Linear behavior</td>
<td>-Needs damping</td>
<td></td>
</tr>
<tr>
<td>Telescoping boom</td>
<td>-Precise</td>
<td>-Not very compact</td>
<td>QuakeSat, REXUS 9, REXUS 11</td>
</tr>
</tbody>
</table>

[17]: Image of antenna deployment with composite boom
[18]: Reference to relaxation effects in tape springs
[19]: Reference to the time and structure on energy storage.
### Table 2.1: Summary of the studied deployment methods

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Pros</th>
<th>Cons</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wire boom</td>
<td>- Lightweight</td>
<td>- No control of the orientation</td>
<td>- DICE</td>
</tr>
<tr>
<td></td>
<td>- Passive</td>
<td>- Needs rotation to deploy</td>
<td>- THEMIS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- MEFISTO-S</td>
</tr>
<tr>
<td>Origami foldable structure</td>
<td>- Compact</td>
<td>- Needs rotation to deploy</td>
<td>- COSMOS-1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- ISAS/JAXA</td>
</tr>
<tr>
<td>Tape spring based structure</td>
<td>- High stiffness</td>
<td>- Non linear behavior</td>
<td>- FedSat</td>
</tr>
<tr>
<td></td>
<td>- Light</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Compact</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.3 Requirements

In order to be able to choose the most appropriate deployment system within the studied ones, a list with all the requirements that the system needs is presented below:

1. **Size of the solar panel surface:**
   
The solar panel system must be able to provide with enough energy to the system while the mission is ongoing. The specification regarding the surface of the panels required was already given. The total surface of the panel must not be smaller than $5120\, \text{cm}^2$.

2. **Size of the mechanism:**
   
The mechanism must fit within the surface of the rover when stowed. It must be placed on top of it and be as low as possible (always reaching the desired area of $5120\, \text{cm}^2$), but also must not have any part sticking out of the platform (being the size of the platform $320\times560\, \text{mm}$, and taking into consideration that other elements are placed on it).

   When deployed, the system must not collide with any other part of the rover in any position.

3. **Main position of the panels:**
   
   As mentioned in previous sections, the rover’s mission will be to the south pole of the moon. Therefore the sun rays will not reach the surface of the moon perpendicularly as happens in the case of missions in the equatorial plane, but in a more tangent to the surface direction. Therefore, the panels must be placed in a vertical plane.

4. **Weight:**
   
   As in all space missions, the weight is an important factor. The system must be as light as possible, but always respecting the resistance and
stiffness needed. There is not a numeric requirement, but this is a fact to take into consideration when designing the individual parts.

5. Spacial restrictions:

It is always very expensive to send any product to space. Therefore the quality of the product must be very good and space-proof. Some parts, such as bearings and motors, will need to be sealed in order not to suffer from the space conditions, while others will not need to and will therefore be optimized for both price and weight.

6. Solar tracking:

The solar panel system must be able to track the sun’s position in order to maximize the energy obtained.

7. Clash-free:

The area covered by the solar panel in each position must be free of obstacles so that the panel do not hit any element or other system when moving. There is also a radiation detector that must not be covered in any moment.
Chapter 3

Implementation

In this chapter, the decision on which system to implement, regarding the previously shown requirements, will be made. The system will also be designed on CAD so that a suitable implementation can be presented.

3.1 System to implement

Within the studied possibilities, the chosen one is the use of a tape spring mast which will pull upwards the rolled solar panels from the top. The solar panels have to be able to rotate with respect to its vertical axis in order to follow the Sun’s position. The analysis done for making the decision is shown below.

The system will be implemented in a lunar rover, and it needs to be able to control the orientation of the panels. The wire boom is therefore not valid for the application, as it is not able to control the orientation of the panels when deployed.

Among the rest of the studied possibilities, other criteria will be examined:

An articulated structure can be used by stacking the panels one on top of the other as shown in Figure 2.1. And in the case of articulated truss structures, some flexible panels can be rolled and pulled up by the tip of the truss. Therefore those two systems could be used for the application.

In the case of inflatable booms, the mast could be used in the same way as explained for the articulated truss structures. Nevertheless, this option has the disadvantage of needing a storage for the gas that is pumped inside the boom to inflate it. Therefore the space needed for storing it is bigger and the articulated truss structure would be preferred in this case.

Regarding torsion springs, there is a need of a lot of space for storing the panels. In the applications in which they are used, the panels are stuck to the
satellite’s surface when they are stored. Nevertheless, in the case of a rover, that space is taken by the wheels or other components, which does not allow this to be the main deployment system.

A telescoping boom would be used in the same way as articulated truss structures. The tip of the mechanism would pull the panels which would be rolled in a stowed position. Nevertheless, this method also takes more space than the truss masts, and truss structures will be therefore preferred.

The origami foldable structure is suitable for satellites where there is enough space around the rotation axis. Nevertheless, in a moon rover there are devices that could be damaged with this kind of deployment. Therefore this system will not be used either.

Lastly, tape spring based structures seem suitable when applied as truss structures, i.e. as mast to pull the tip and “unroll” the solar panels. This method could therefore also be used, and it has the advantage of being lighter than the truss structures and than the articulated structures.

The system to be implemented is therefore a tape spring mast that pulls from the tip of a rolled solar panel.

Even if there is no vertical restriction for the size of the panels, a lower size will be more appropriate, as it will create lower torques in the base. Therefore, a lateral deployment system will be included. This way, the stowed system will fit inside the rover, but when deployed it will be wider. This deployment can be achieved via torsion springs or linear motion. This will be studied in the subsection Solar panel lateral deployment.

3.2 Solar panel storage unit

As mentioned in previous sections, the solar panels will be rolled when stowed, and the system must be able to rotate. This system will be developed in the upcoming subsections.

3.2.1 Base plate

The first part is the plate where the solar panels will be attached. This will be the support for the shaft in which the solar panels are rolled. The designed part is shown in Figure 3.1 and its properties are summarized in Table 3.1.

This plate is symmetric and made out of aluminum sheet metal of 3mm thickness. The coupling with the rotatory system is placed on the middle, and two holes (marked in red) are made for parallel pins to ensure that the

---

The properties shown for all the elements on each table are basic sizes. The details can be seen in the drawings in the annex.
CHAPTER 3. IMPLEMENTATION

Figure 3.1: Base plate of the solar deployment system

The orientation of the plate is the correct one with respect to the motor’s shaft (the tolerances for this holes will be tighter than the average tolerances).

The four holes in the corners are made in order to attach the shaft, and the bigger holes throughout the plate are designed in order to reduce the weight of the piece.

In order to ensure that the plate is appropriate for the application, a simulation has been done, with a total force of 20N, which is greater than the expected real force. The maximum displacement in this conditions is around 2mm, which is acceptable taking into consideration that the force will be smaller than the one studied. The results of the simulation are shown in Figure 3.2.

Figure 3.2: Simulation of the base plate
### Table 3.1: Characteristics of the base plate

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>100 mm</td>
</tr>
<tr>
<td>Length</td>
<td>381 mm</td>
</tr>
<tr>
<td>Thickness</td>
<td>3 mm</td>
</tr>
<tr>
<td>Material</td>
<td>Aluminium</td>
</tr>
<tr>
<td>Weight</td>
<td>169.26 g</td>
</tr>
</tbody>
</table>

#### 3.2.2 Shaft

As specified, the solar panels will be rolled when stowed. The selected object to roll the panels on will be a hollow shaft as the one represented in [Figure 3.3](#) with threaded ends.

The fact that it is hollow, apart from reducing weight, will allow the cables from the solar panels to go inside and therefore not get tangled. The ends will be threaded just for testing purposes. During the test phase, several rolling and unrolling phases will be made. The threads in the ends will allow the rolling manually from the outside by attaching some handles. Nevertheless, as the system will be deployed just once when implemented in the real application, these threads will not be used for any task.

![Figure 3.3: Shaft](#)

The diameter is different on the edges in order to be able to fix it with the rest of the system. The coupling with the plate is shown in [Figure 3.4](#). As seen, the edges of the shaft will be connected to two radial bearings with their respective cases. The cases will be attached to the base plate with the structure shown in [Figure 3.5](#). This structure is made of sheet metal and is only necessary for giving some extra height to the shaft, as the bearing support is not tall enough to let space for the panels when rolled.

#### 3.2.3 Motor and coupling

In order to make the system to rotate along its vertical axis, a motor is needed. The motor has been chosen taking into consideration the system’s
The motor needs to be attached to the Rover, and it’s shaft will be connected to the plate with a bearing system. The coupling system is shown in Figure 3.6 and the separate pieces in Figure 3.7. The basic characteristics of the pieces can be found in Tables 3.3 to 3.6.
CHAPTER 3. IMPLEMENTATION

Figure 3.6: Motor and plate coupling

(a) Assembly
(b) Exploded view

Figure 3.7: Motor and plate coupling

(a) Outer bearing case
(b) Inner top bearing case
(c) Inner bottom bearing case
(d) Motor holder
### Table 3.3: Characteristics of the outer bearing case

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>60x60 mm</td>
</tr>
<tr>
<td>Thickness</td>
<td>7.8 mm</td>
</tr>
<tr>
<td>Big</td>
<td>52 mm</td>
</tr>
<tr>
<td>Small</td>
<td>47 mm</td>
</tr>
<tr>
<td>Material</td>
<td>Aluminium</td>
</tr>
<tr>
<td>Weight</td>
<td>32.72 g</td>
</tr>
</tbody>
</table>

### Table 3.4: Characteristics of the top inner bearing case

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Height big</td>
<td>2 mm</td>
</tr>
<tr>
<td>Height medium</td>
<td>8 mm</td>
</tr>
<tr>
<td>Height small</td>
<td>20.6 mm</td>
</tr>
<tr>
<td>Big</td>
<td>60 mm</td>
</tr>
<tr>
<td>Medium</td>
<td>31 mm</td>
</tr>
<tr>
<td>Small</td>
<td>25 mm</td>
</tr>
<tr>
<td>Material</td>
<td>Aluminium</td>
</tr>
<tr>
<td>Weight</td>
<td>35.01 g</td>
</tr>
</tbody>
</table>

### Table 3.5: Characteristics of the bottom inner bearing case

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big</td>
<td>31 mm</td>
</tr>
<tr>
<td>Small</td>
<td>10.4 mm</td>
</tr>
<tr>
<td>Material</td>
<td>Aluminium</td>
</tr>
<tr>
<td>Weight</td>
<td>10.22 g</td>
</tr>
</tbody>
</table>

### Table 3.6: Characteristics of the motor holder

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness</td>
<td>2 mm</td>
</tr>
<tr>
<td>Height</td>
<td>14 mm</td>
</tr>
<tr>
<td>Width</td>
<td>80 mm</td>
</tr>
<tr>
<td>Material</td>
<td>Aluminium</td>
</tr>
<tr>
<td>Weight</td>
<td>12.10 g</td>
</tr>
</tbody>
</table>

### 3.2.4 Solar cells’ support

As specified in the previous sections, the solar panels will be rolled in a shaft until deployment. Therefore, solar cells that stick to a flexible thin element will be used. The element that will support the solar cells will be a flexible sheet of carbon fiber (Figure 3.8), as it is available in very thin configurations and with a smooth surface, being at the same time very light. Its properties are shown
in Table 3.7. The smooth surface of the sheet will allow the use of glue to stick the solar cells. It will also prevent from scratching the cells that are rolled on top of it.

![Flexible carbon fiber sheet](image)

Figure 3.8: Flexible carbon fiber sheet used as support for the panels

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness</td>
<td>0.25 mm</td>
</tr>
<tr>
<td>Surface</td>
<td>5120 cm²</td>
</tr>
<tr>
<td>Material</td>
<td>Flexible carbon fibre</td>
</tr>
<tr>
<td>Weight</td>
<td>224 g</td>
</tr>
</tbody>
</table>

Table 3.7: Characteristics of the cells’ support

### 3.2.5 Complete base system

The different parts of the solar panel support system are assembled using rivets where it is possible, in order to prevent the parts from getting loose with the vibrations. For the parts where rivets cannot be used, the fixture is made using threaded holes, screws, nuts and washers. The whole assembly (not including the representation of the solar cell’s support (as it would make the figure unclear) is shown in Figure 3.9.

Note that, the black carbon fiber square shown in the CAD represents the surface of the Rover, where the system will be attached. For illustrative purposes, this part has been cut in order to be able to show the relative positions of each component with the Rover’s surface, without completely covering the part "inside" the Rover.

### 3.3 Solar panel lateral deployment

As specified previously, reducing the height of the system will prevent from having high torques in the base of the system. Therefore, the panel is made wider using flaps to deploy laterally.
3.3.1 Base idea

The idea of deploying in two phases is born when the area of the panels is compared to the size of the Rover. In case of building a solar panel that is just a rectangle, with the panel surface given, and that can fit without hitting any other part, the height of the panel would be very high. Therefore, including a lateral deployment, the panel can be shorter and wider, but in the stowed position it still fits inside the Rover’s plant.

As mentioned previously, this deployment phase can be accomplished either by using torsion springs and a rotation movement of the lateral panels or linear springs (causing linear motion of the lateral parts of the panels).

In order to make a decision on this topic, the pros and cons of each option have been studied and are shown in Table 3.8.

<table>
<thead>
<tr>
<th>Mechanism</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>- Not covering surface of the central panel</td>
<td>- No possibility of attaching the lower part of the lateral panel</td>
</tr>
<tr>
<td></td>
<td>- Simple mechanism</td>
<td></td>
</tr>
<tr>
<td>Rotational</td>
<td>- Simple mechanism</td>
<td>- In case of failure, the whole surface of panels will be covered</td>
</tr>
<tr>
<td></td>
<td>- Used for other elements of the Rover</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Possibility of attaching the lower part of the lateral panel</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.8: Pros and cons of each method for lateral deployment
3.3.2 Design of the system

Due to the con against the rotational option, the linear one has been developed instead. In order to do this, the panels must be attached from the top. The attachment will be done by a piece of metal which will be pulled upwards when deploying.

In order to do the deployment, a Hold Down Release Mechanism (HDRM) will be used. The upper metal will be provided with guides so that the sliding can be done to push out the flaps, and there will be magnets so that once the flaps are released, they do not slide in again due to inertias.

The HDRM are easy to find from commercial brands, some of them specialized in space applications (such as NEA electronics). Commercial mechanisms will reduce the price, keeping the quality and reliability of the product.

Therefore, the only piece to design for this task is the upper holder of each section of panel. This has been designed in CAD and can be seen in Figure 3.10. Its main characteristics are shown in Tables 3.9 and 3.10.

![Figure 3.10: Upper holder for the panels](image)

(a) Holder for the central panel
(b) Holder for the lateral panels

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness</td>
<td>2 mm</td>
</tr>
<tr>
<td>Length</td>
<td>352 mm</td>
</tr>
<tr>
<td>Width</td>
<td>65 mm</td>
</tr>
<tr>
<td>Height (max)</td>
<td>45 mm</td>
</tr>
<tr>
<td>Material</td>
<td>Aluminium</td>
</tr>
<tr>
<td>Weight</td>
<td>153.67 g</td>
</tr>
</tbody>
</table>

Table 3.9: Characteristics of the upper holder for the central panel

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thickness</td>
<td>2 mm</td>
</tr>
<tr>
<td>Length (max)</td>
<td>340 mm</td>
</tr>
<tr>
<td>Width (max)</td>
<td>12 mm</td>
</tr>
<tr>
<td>Height (max)</td>
<td>10 mm</td>
</tr>
<tr>
<td>Material</td>
<td>Aluminium</td>
</tr>
<tr>
<td>Weight</td>
<td>28.26 g</td>
</tr>
</tbody>
</table>

Table 3.10: Characteristics of the upper holder for the lateral panels

The attachment of the two parts will be done by using sliders, and the
deployment will be produced by springs and a commercial HDRM system. The stowed and deployed versions of it can be seen in Figure 3.11.

![Stowed system](image1)

(a) Stowed system

![Holder for the lateral panels](image2)

(b) Holder for the lateral panels

Figure 3.11: Lateral deployment phase

### 3.4 Boom system

In this section, the mechanism to pull the panels up will be explained.

#### 3.4.1 Design

The boom has to be designed in relation with the size of the panels, as it needs to reach (at least) the height for the solar panels to have the required area. The boom also needs to be as compact as possible in its stowed configuration, and therefore its length should be adjusted to the minimum required height.

It is also necessary to create a system that deploys the boom, as it is not necessarily self-deployable. Having a motor to deploy the boom would also make it possible to have adjustable height of the mast.

The whole system needs to rotate with the panels, and therefore it needs to be mounted on the same surface that the panels are mounted on.
3.4.2 Results

Composite booms have been recently used in space applications, and therefore, there are companies that commercialize them and are specialized on building these systems. The main two possibilities were:

- Buy the boom and design the system to deploy it
- Buy the complete system

As happened with the HDRM, buying an already built and tested mechanism will reduce costs of the project, increasing the reliability of the system. This happens because the companies that commercialize this systems are specialized on them and provide some given guaranties. Building the system from scratch would mean that the individual elements have to be space-rated, and that the system has to be tested afterwards. This would increase the cost of the system, as the tests for space applications are expensive.

The boom system will therefore be bought. Within the providers, the chosen system is shown in Figure 3.12a. The system has been drawn in a simplified manner in CAD (Figure 3.12b) so that the space for it could be taken into consideration, making also a more completed CAD model of the whole system.

![Figure 3.12: Boom deployment system](image)
3.5 Complete system

Once the individual parts of the system have been designed, the system needs to be assembled. This is a way of checking that all the parts work well together, and with the rest of the Rover.

3.5.1 Coupling between the parts

The coupling of some of the parts has already been explained in previous subsections. Therefore, this analysis will be done by parts.

- The photovoltaic cells will be attached just to the panels support (subsection 3.2.4). This attachment will be done with special glue suitable for the given application. This glue needs to be suitable for carbon fiber, and must also be suitable for space applications.

- The cells’ support will be attached on one end to the holder designed in subsection 3.3.2 by rivets, which will not get loose with vibrations. In case of the lateral flaps, this will be the only attachment. For the central panels, the support will also be attached to the shaft (subsection 3.2.2) using rivets. This is possible, given that the shaft is hollow.

- The holders (subsection 3.3.2) of the lateral panels will be attached to the central one, as explained previously, while the central panel’s holder will be attached to the boom using two rivets. Two rivets will prevent from rotation in case that the center of mass of the panels is not perfectly centered (or changes due to external factors).

- The shaft will be attached to the rotatory base by the use of radial bearings as shown in Figure 3.4. The bearings on the sides will allow the shaft to freely rotate allowing the panel to unroll.

- The base (subsection 3.2.1) will be attached to the motor and the Rover with the system shown in Figure 3.6. This system includes diagonal bearings so that the rotation of the motor is transmitted properly to the rest of the system, without big energy loses. The diagonal bearing can also account for the possible radial loads, which are unwanted and might break the motor otherwise.

- Finally, the boom will be attached to the base (subsection 3.2.1) by its case. The case of the boom is not represented in the CAD models, as it is a commercial element instead of an own-designed one.
Chapter 4

Results and Conclusions

To finish with the hardware part of the project, the results of the design phase are shown, along with some conclusions.

The system has been developed and assembled in CAD so that the deployment sequence could be simulated. The sequence is pictured in Figure 4.1.

Figure 4.1: Deployment sequence of the solar panel system
In the first step, the system is fully stowed. The space that it takes is small regarding the total surface of the solar panels. The panels are rolled in the shaft, and the width that it takes is small enough to fit into the horizontal surface of the Rover, also not interfering with any other element in the stowed position.

In the second step, the first part of the deployment is completed. The central panel is in vertical position and the boom is fully deployed. In this step, the Rover is already able to obtain energy. Therefore, even if the third phase of the deployment is not completed, the system would work (but obtaining less energy than what was planned, as the effective surface would not be the designed one).

Lastly, when the third phase of the deployment is fulfilled, the effective surface of the panel is the designed one, the panels are fully deployed, and they should be locked in vertical position (and the flaps locked with respect to the central panel).

In order to complete the project regarding the mechanical part, the system should be built and assembled into the vehicle (or a prototype of it). The system can be easily modified, as it is in some aspects modular. Some examples of the modifications that can be easily done are the following:

- The flaps can be deleted in case there is no need of a big surface. This modification would not affect any of the other elements of the system, as they are already an addition.

- The deployment of the flaps could be done via rotation in case it is preferred. The only elements that should be modified are the holders of the panels on top, as they would not need a sliding guide anymore.

- The size of the panels can be easily modified just changing the height of the boom (or configuring the motor that deploys it so that it unrolls less boom).

- The width of the system can be made smaller (or bigger) in case that it needs to be implemented in a smaller space.
Part II

Control
Chapter 5

Introduction

To begin with the control part, a brief introduction is made. First, the background and aim for the project will be analyzed. The state of the art will be studied in order to find a suitable solution to the given problem. To finish with this chapter, a brief list of the requirements needed is presented.

5.1 Background

Since the previous Rover did not have solar panels, in the beginning of the project there was no specific control approach. Nevertheless, a simple and logical idea would be to control the solar panels in a way that they track the Sun rays at each moment in order to make the charging process faster.

The robot will also go to the shadowed regions of the Moon (such as deep craters), which will imply that there will be moments in which no energy will be produced. For these situations, in which no energy can be obtained, the panels should not rotate, as it would only spend energy without any generation. This fact will have to be taken into consideration when controlling the complete system, in order to allow the Rover to get energy before the batteries are emptied.

5.2 Aim

The aim of the project, regarding the control part, will be the efficient orientation of the solar panels once they have been deployed. The system should be able to track the Sun’s position with respect to the vehicle’s orientation, making a good use of the energy produced by itself.

Therefore, the movement of the panels should be optimized according to the
trajectory that is planned for the vehicle, in order not to move in an inefficient way the panels when it is not necessary.

5.3 State of the art in solar tracking systems

In order to be able to choose the appropriate control system for the solar panels, a study of the controllers previously used in solar applications has been done. The aim of the project will be obtaining as much energy as possible. For this task, a first classification on the control strategies has been done according to [21].

The first group of strategies contains the low-level controllers. These controllers focus on driving a motor to a desired position, not taking into account the rest of the system. This could, for example, represent the controller of the motor that drives a wheel of a vehicle. The controller would not be aware of the rest of the system, but just on the correct movement of the wheel.

The second group would be the high-level controllers. These strategies are usually combined with low-level controllers. This way, and following the previous example, the high level controller would be in charge of driving the whole vehicle through a trajectory by calculating how to move each wheel of the vehicle. Then the controller of each wheel would be in charge of making this movement happen.

In the following, the different strategies seen in applications or proposed for them will be examined.

Manual tracking

For private installations there are many cases in which no controller is applied. The panels are fixed and can be manually moved and re-oriented depending on the time of the year. Just by the movement of the panels twice a year, the efficiency increases from 71.1% to 75.2% for a location at 40° latitude [22]. This value can be increased by re-orienting the panels more frequently.

Private systems are usually used as a second source of energy. Therefore, in many cases, the owner does not feel the need of installing an automated system, as it would increase the cost of the installation and the reward would not be enough to compensate for this.

Automated tracking in private systems

In some cases, an automated tracker is used in private installations in order to increase the efficiency. The tracking can be done in either one or multiple axes [23].
When the tracking is done for just one axis, the aim is to track the azimuth, i.e. the movement of the Sun across the sky for one day. If the panels are slightly tilted, facing to the south, the annual power production can be increased. When two rotation axes are used to track the Sun, the panels are able to follow the Sun not only in a daily basis, but also throughout the year. The angles controlled in this case are the azimuth and the tilt.

A scheme representing the types of tracking that can be implemented for this application in more detail is shown in Figure 5.1. The controllers used for these case can be either non-linear (such as on-off regulators) or linear. For this last case, P, PI or PID controllers can be used, as well as other more complicated control structures.

These trackers can be either open-loop or closed-loop. The open-loop systems use mathematical equations in order to guess where the Sun is at a given time in a given place. For the case of closed-loop systems, sensors are used in order to find the Sun’s position, and the panels are controlled via feedback.

An example of a closed-loop implementation can be found in [25]. Firstly, a system with two rotation axes was proposed. In order to build the system, a dual-axis motor was used (instead of the classical implementation using two motors), and the system was equipped with four light sensors in order to be able to find the appropriate orientation. This system was built and tested against a fixed solar panel in order to check the improvement on the efficiency.

Figure 5.1: Scheme of the tracking strategies implemented for simple systems [24]
CHAPTER 5. INTRODUCTION

Tracking in solar power plants

When it comes to solar power plants, where the main goal is to maximize the energy production, more complex systems might be used. Nevertheless, simple controllers similar to the ones described in the previous section can be used.

Photovoltaic power plants

One of the most important factors, in case this technology is used, is the average daily intensity of solar radiation that the panels receive [24]. This factor depends on the season, the weather conditions and location of the Sun in the sky, among others. The orientation of the panels is a very important factor, as the angle formed between its normal and the direction of the Sun is critical.

In order to maximize the energy production, several methods have been studied and proposed. These methods range from simple controllers based on the theoretical position of the Sun, considering location of the plant, hour of the day, season, etc. to more complex ones such as the use of neural networks or genetic algorithms.

The systems used can be either active or passive. The first ones are more complex than the second group. Nevertheless, they are also more efficient and more widely used. Active systems are usually based on the use of sensors and use feedback control to follow the position of the Sun. Even if the automated systems are usually more efficient than the fixed ones, for hard weather conditions, the system might fail and be less efficient than a fixed one.

Thermal power plants

Thermal power plants are based on the use of mirrors that concentrate the Sun rays (and therefore the heat) into one point in order to heat a liquid and produce energy in the same way other power plants work. In these kind of power plants, there are several levels of control, as not only the orientation of
the mirrors is to be adjusted\cite{26}. These levels are:

- Control of the movement of the mirrors
- Control of the thermal process variables
- Plant control aspects
- Grid integration

In the case that the only goal were obtaining as much energy as possible, then just some of the levels of control would be useful. Nevertheless, the power plant needs to work around a setpoint, and therefore some communication between the levels is needed.

The lower level controllers are mainly open-loop, but they have some closed-loop adjustments according to the working setpoint. For the overall control of the energy obtained, many different techniques, ranging from classical PID control to complex neural network controllers, have been used.

According to\cite{26}, several different implementations have been tested, finding that the most successful implementation for the ACUREX plant consists on a gain scheduling Model Predictive Control (MPC). MPC will be briefly introduced later on.

Solar panels in space

In order to be able to choose the best option for the control of the panels of the vehicle, a brief research work has been done with regards to the control of the solar panel systems used in spacecrafts, satellites or vehicles.

Orbiting satellites

The most commonly source of energy used in satellites is solar photovoltaic. The first small satellites launched that were solar-powered had the panels surrounding the body, and fixed. An example of this is the Vanguard 1, the fourth artificial Earth-orbiting successful satellite\cite{27}. The solar system powered just some of the subsystems of the satellite, while the main propulsion system was powered by batteries\cite{27}.

Nevertheless, nowadays most of the satellites are equipped with solar trackers in order to maximize the energy obtained. An example of this is shown in Figure 5.3\cite{28}. The ISS (International Space Station) is powered by solar photovoltaic technology and some batteries that are charged when the ISS is not shadowed by the Earth\cite{28}. Therefore, while there is light available, the system tries to get as much energy as possible in order to charge the batteries.
Planetary vehicles

For the last years, most of the planetary (or lunar) missions had as aim to reach the equatorial plane of the visited places. Therefore, the panels were usually covering the top of the vehicle, as the solar rays usually reached the surface of the planet perpendicularly to the surface.

These kind of systems have an implementation similar to Earth-applications’. The panels might be movable, in order to increase the efficiency, but in case they are not, the loss of efficiency is not as critical as in vertical systems. An example of this implementation can be seen in Figure 5.4.

With the change of destination of the vehicles, the control of the panels becomes more important. This happens because when making the panels vertical (which is necessary in order to be as perpendicular as possible to the Sun rays), the movement of the vehicle might cause that the panels face in a complete opposite direction (which would not happen with horizontal panels, as the vehicle will never be upside down).
Advanced controllers

Apart from the previously mentioned controllers used, two advanced techniques have been briefly studied in order to find the best solution for the problem given. These two techniques are MPC and Adaptive Control (AC).

Model predictive control

MPC is a technique that has already been used in some solar thermal power plants in order to manage the functioning of the whole plant, while the control of the mirrors is done via a simpler controller. Therefore the MPC controller is used as a ”brain” in order to maintain an appropriate working setpoint.

MPC control is based on having a model of the system to be controlled, in order to make decisions on what to do so that the system gets to a desired state. With this method, constraints can be set, and some variables can be given more importance than others. This way, the behavior of the system can be adjusted and optimized.

The main advantages of MPC are its straightforward formulation, the ease of handling constraints and maintenance, the fact that the tuning parameters are easily understandable as physical entities (prediction horizon) and the fact that its development time is usually shorter than for other advanced control methods [31].

Adaptive control

The last control technique to be briefly studied is Adaptive Control. This technique is widely used to deal with unknown uncertainties. These controllers adapt themselves in order to handle the uncertainties of the model. It has been used in various real world applications [32].

This control technique is used mainly in systems with varying dynamics. An example of this is autonomous vehicles, which will spend fuel while moving, and therefore lose weight during operation. Other examples are tools that will suffer from wear and tear, and therefore the tip of the tool will not be always at the same distance form the holding part. The controller can be designed using various methods, such as gain scheduling and self-tuning regulators [33].

5.4 Requirements

In order for the system to be suitable for the given application, a set of requirements of the system has been defined.
1. Energy trade-off:
   The controller must find an appropriate trade-off between obtained energy and energy spent to move the panels. Note that, even if the energy obtained by the panels will be used for all the subsystems of the vehicle, only the energy spent by the movement of the panels will be taken into consideration when solving the optimization.

2. Movement of the Rover:
   The system must account for the vehicle’s orientation as it moves, having it into consideration when calculating the optimal orientation for the panel throughout time.

3. Zero energy places:
   The system should not move when the vehicle is in a shadowed place, even if the energy trade-off says so (considering just the relative position of the Sun). Even if the panel is pointing at the Sun’s direction, if the Sun is hidden, the produced energy will not be the same as the estimated one.

4. Fully charged battery:
   As in the previous case, when the batteries are fully charged and there is no possibility of storing more energy, the panels should not move.
Chapter 6

Implementation

In this chapter, the mathematical description of the system will be presented, starting with the decision on which system will be implemented regarding the requirements, to finally obtain a suitable controller, obtaining also the necessary models for it.

6.1 Algorithm to implement

Within the studied possibilities, the approach that will be applied for the project will be Model Predictive Control (MPC). This control technique is suitable for the project, as it will be able to make a decision on when to move the panels and when not to move them (as well as how much to move them), according to the energy efficiency criteria. This controller will need to have a model of both energy consumption and energy generation in order to take the right decisions, as well as knowledge of the orientation of the Sun with respect to the Rover at each timestep.

6.2 Base idea

The base idea followed in order to find a good implementation of the algorithm is shown in Figure 6.1, where:

- **Ideal Panel Trajectory** represents the sequence of orientations that the panels should follow in order to be perpendicular to the Sun’s rays at every timestep. Nevertheless, this might not be the optimal trajectory regarding power consumption, and it can even be an impossible trajectory to follow by the system, and therefore other elements are added to the controller.
• *Current position* represents the orientation of the panels at the current
timestep\(^1\). This signal is the same as *Panel’s position*. This signal will
act as the feedback of the system, allowing it to react against possible
disturbances or errors in the modelled system.

• \( u \) represents the input to the system.

• The *OPTIMIZATION* block represents the controller. At each timestep,
the controller solves an optimal control problem for \( n \) timesteps and out-
puts only the first value of the solution’s sequence. The optimization
problem to solve is the maximization of the *Energy Balance*. Inside this
block, the elements are the following:

  – *Generated Energy* includes a model of the energy that is generated for
    a specific trajectory (*possible trajectories*), knowing what the *Ideal
    Panel Trajectory* is.

  – *Consumed Energy* has a model of the motor, and gives information
    about the amount of energy that is spent in order to follow a specific
    trajectory (*possible trajectories*) knowing the *Current position*.

  – The subtraction of the generated and consumed energy gives the *En-
    ergy Balance*, which will be maximized when solving the optimization
    problem for the *possible trajectories*.

• Lastly, the block *Motor* represents the actuator of the system. Given an
input, it will rotate to orientate the panels.

\[ \text{OPTIMIZATION} \]
\[ \text{Maximize Energy Balance w.r.t. the possible trajectories} \]
\[ \text{Ideal Panel trajectory} \]
\[ \text{Generated Energy} \]
\[ \text{Current position} \]
\[ \text{Consumed Energy} \]
\[ \text{Motor} \]
\[ \text{Panel’s position} \]

Figure 6.1: Overview of the controller’s scheme

The sketched algorithm covers the first two requirements, while the third
and fourth ones must be implemented externally. The three main options for
solving the third problem are:

1. Have a light sensor which triggers a flag in the algorithm which avoids
using the controller when the vehicle is in a dark location.

\(^1\)Note that both *Ideal Panel Trajectory* and *Current position* are inputs to the optimization
block, and can be used by all the blocks inside it.

41
2. Keep track of the energy that is being obtained in the last period of time (length can be tweaked) and compare it to the obtained energy that the controller predicted for that same period of time. Trigger the same flag as previously if there is a big difference between both.

3. If the places where there is not going to be light are known beforehand, trigger the flag externally when the localization system of the Rover states that the vehicle is in a dark location.

For the case of the fourth requirement, a flag should also be used in the same manner as in the previous case. Nevertheless, this flag can only be triggered by an external sensor. This sensor should measure the amount of energy stored in the battery, and when a certain level is reached, the flag that switches off the control should be turned on.

### 6.3 Models

In this section, each element of the controller’s scheme shown in Figure 6.1 will be studied. Note that, inside the **OPTIMIZATION** block, the solution is found numerically, and therefore, the algorithms might differ from the ones shown in this section. In this section, two algorithms (one for each block) that are suitable to find the Energy Balance from the given inputs (i.e. **Current Orientation** and **Ideal Panel Trajectory**), along with the intermediate signal (**Sample \( u_n \)**, considered internal for the optimization) will be sketched.

#### 6.3.1 Ideal Panel Trajectory

In order to get the so called **Ideal Panel Trajectory**, the two involved components are the trajectory of the Sun across the sky and the trajectory of the vehicle on its path through the Moon’s surface. Both of them will be taken into account, so that the **Ideal Panel Trajectory** can be computed according to a scheme similar to Figure 6.2. The elements of this scheme will be studied in the following points.

![Figure 6.2: Computation of the orientation of the rover with respect to the Sun](image)

**Figure 6.2: Computation of the orientation of the rover with respect to the Sun**

**Sun’s trajectory**

The first component to be taken into consideration will be the trajectory of the Sun across the sky of the Moon. This movement will be modelled using
a coordinate system centered on the South pole of the Moon. According to [33], in the South pole of the Moon, the Sun follows a circular trajectory, very close to the horizon with almost no vertical variations. The Sun would take approximately 29.5 Earth days to fulfill a complete cycle, i.e. to return to its initial position. The angular velocity and position with respect to time of the Sun will therefore be:

\[ \omega(t) = \frac{1 \text{ cycle}}{29.5 \text{ days}} = 2.4652 \times 10^{-6} \text{ rad/s} \] (6.1)

\[ \varphi(t) = \omega \cdot (t - t_0) + \varphi_0 = \omega \cdot t \] (6.2)

(Assume \( \varphi_0 = 0 \) and \( t_0 = 0 \))

Figure 6.3: Trajectory of the Sun w.r.t the South pole of the Moon for one cycle

**Desired trajectory of the rover**

The *Desired trajectory of the Rover* signal represents a planned trajectory for the vehicle. This planned trajectory will be used in order to compute the angle of the Rover with respect to the Moon, given a parametric equation of the Rover’s trajectory and its speed. An example of a valid trajectory, which will be used for the last simulations, is shown in Figure 6.3. The trajectory must be given before the movement starts, in order to make the controller able to plan the rotation of the panels in advance.

**Rover’s real trajectory**

This signal consists in a sequence of orientations that the vehicle has with respect to the coordinate system fixed on the south pole of the Moon. In order to get this orientations sequence, a conversion of the *Desired trajectory of the*
Rover is needed. This conversion will be done in block *Adjustments of the Rover*.

**Adjustments of the Rover**

In order to do the conversion between the two previous signals, this block is used. The first approach will be using the tangent to the trajectory in each point as the real orientation of the Rover. In case that the controller with this approach works properly, a more detailed model of the vehicle’s orientation might be used in order to improve the performance. Using this more detailed model, the orientation of the rover would not be considered as the tangent to its planned trajectory, but as another value close to it.

In order to optimize the problem, instead of computing the orientation for many points, the amount of points for which the orientation will be computed is reduced, always taking an amount that reflects the evolution of the orientation properly. Once the orientation is known for several time points, a polynomial will be fitted to the data (using several batches according to the timestep, so that the polynomial is simple) so that the controller has access to a continuous function which represents the approximate orientation of the Rover with respect to the Moon.

**Real orientation calculator**

The two previous trajectories will be merged in order to create an *Ideal Panel Trajectory*, which will be expressed in the coordinate system of the Rover. This trajectory will be a sequence of orientations that the Sun has with respect to
CHAPTER 6. IMPLEMENTATION

The Rover at each timestep.

The first thing to consider is that the distance between the systems is much smaller than the distance to the Sun, and therefore the displacements of one system with respect to the other will not make a difference in terms of orientation of the systems with respect to the Sun. The change of coordinates can therefore be simplified to just a rotation, taking into consideration just the change of orientation of the Rover’s system, as shown in Figure 6.5.

![Figure 6.5: Change of coordinates scheme](image)

The Ideal Panel Trajectory will therefore be a sequence of angles that represent the orientation of the Sun with respect to the vehicle at each timestep. These values are given by Equation 6.3.

\[ \varphi_{\text{Sun,Rover}} = \varphi_{\text{Sun,Moon}} - \varphi_{\text{Rover,Moon}} \] (6.3)

6.3.2 Generated Energy

The energy that is generated by the panels depends on several factors, such as the angle with which the solar rays reach the surface of the panel, the type of panel, the efficiency of the panel, the amount of dust that is accumulated on the panel’s surface (whose effect has been studied in [35]), to name a few. The amount, size and type of dust will not be taken into consideration in the problem, as it is an unknown parameter, and it will affect equally in any configuration.

The agent that will be determinant for the system to optimize the power usage is the angle between the line that goes from the panel’s surface to the Sun and the orthogonal line to the panel’s surface. This angle will be from now on denoted as the angle of incidence, and its optimal value is 0° for maximum power generation. Therefore, in order to compute the energy obtained when having a sample input sequence, the diagram shown in Figure 6.6 will be used. The inputs to the calculator are the Ideal Panel Trajectory and the signal called Sample \( u_n \).

---

4The problem will be considered as a 2 dimensional problem, as there is just one degree of freedom for the control of the solar panel.
The effect of the angle of incidence has been studied in [36], where an experimental approach was used to find out the loss of efficiency of the solar panels for different angles of incidence (Equation 6.4, Figure 6.7),

$$D = f(\varphi_{\text{incid.}}) = 1.41669574 \cdot 10^{-9} \varphi_{\text{incid.}}^6 - 3.59548024 \cdot 10^{-9} \varphi_{\text{incid.}}^5 - 5.57767010 \cdot 10^{-6} \varphi_{\text{incid.}}^4 + 1.64493758 \cdot 10^{-5} \varphi_{\text{incid.}}^3 - 1.724028491 \cdot 10^{-4} \varphi_{\text{incid.}}^2 - 7.11435953 \cdot 10^{-3} \varphi_{\text{incid.}} + 99.37322857 \quad (6.4)$$

where D represents the percentage of power that is obtained and $\varphi_{\text{incid.}}$ is the angle of incidence.

Note that the efficiency is not reaching 100% for any angle. This happens because it is calculated experimentally for several tilt values and then fitted into a smooth curve. When fitting into a curve, given that the polynomial degree chosen is lower than the number of datapoints minus one, there is an error. This error is minimized, but its effect is showing a curve which does not reach 100%. 
The produced energy will therefore be:

\[ E_{\text{produced}} = g(\phi_{\text{incid.}}) = \int_{t}^{t+t_s} P_{\text{max}} f(\phi_{\text{incid.}}) dt \quad (6.5) \]

where \( P_{\text{max}} \) represents the peak value of power that can be obtained by the panels, and it is a parameter that is given by the manufacturer. Note that \( \phi_{\text{incid.}} = \phi_{\text{Panel,Rover}} - \phi_{\text{Sun,Rover}} \).

In many cases, the maximum power obtained by the solar panels is not the theoretical one, but a smaller value. Therefore, the value of \( E_{\text{produced}} \) might be multiplied by a factor in order to account for the possible variations.

### 6.3.3 Consumed Energy

The system needs an estimation of the energy spent to move the panels so that it can maximize the Energy Balance. The only considered sink of energy in the system is the motor that rotates the panels, as the Rover’s power usage is not dependent on the generated energy, but on the available energy on the battery. Therefore, a simple scheme of the computation of the consumed energy can be sketched as in Figure 6.8, where the sub-index \( n \) represents a sequence of values of size \( N \) (length of the prediction horizon). The only input needed is therefore the sequence of inputs for which the consumed energy is to be known.

![Figure 6.8: Computation of the energy consumed when having a specific sequence of inputs](image)

As shown, a model of the motor is necessary in order to predict the energy that is used. This model will be presented in the following sections.

### 6.3.4 Motor model

The controller needs a model of the motor in order to estimate the energy consumed when a specific sequence of input voltages is applied (Sample \( u_{\text{in}} \)).

The motor system is composed by the motor itself, the transmission and the panel (load). Therefore, the inertias for all the components must be taken into account.

\footnote{Note that, in theory, the function is symmetric with respect to the vertical axis, and therefore \( f(\phi_{\text{Sun,Rover}} - \phi_{\text{Panel,Rover}}) = f(\phi_{\text{Panel,Rover}} - \phi_{\text{Sun,Rover}}) \).}

\footnote{Note that, this also corresponds to the block called Motor from Figure 6.1 when simulating. Otherwise, it represents a real actuator, but the model is still necessary for the estimation of the consumed energy.}
consideration when building the model. In order to build a mathematical model of the behavior of the system, an electric model, as well as a mechanical model, must be found.

Electrical model

The electrical model of the motor can be represented as shown in Figure 6.9. The parameters can be found in the datasheet of the motor [37], so that everything is known ($U_i$ represents the input voltage and $\dot{\phi}$ the angular velocity of the output shaft).

\[ E = K_{Emf} \cdot \dot{\phi} \]

Figure 6.9: Electrical model of the motor

From the figure, the differential equation that rules the behavior of the system can be found:

\[ U_i = U_L + U_R + E = L \frac{di}{dt} + Ri + K_{Emf} \dot{\phi} \]
\[ \Rightarrow \frac{di}{dt} = \frac{1}{L} (U_i - K_{Emf} \dot{\phi} - Ri) \quad (6.6) \]

The equation that links both mechanical and electrical models represents the fact that the torque applied is proportional to the current:

\[ T_m = K_m i \quad (6.7) \]

where $R$ is the electrical resistance of the motor, $L$ the inductance, $K_{emf}$ the speed constant, and $K_m$ the torque constant. All of these parameters are given on the motor’s datasheet.

Mechanical model

In order to simplify the task, the model has been built by breaking down the system into subsets of components, as shown in Figure 6.10. Therefore, the effect of the inertia of each subset will be taken into consideration in order to find a system that is as representative as possible of the real system. The shafts will in all cases considered to be rigid, as in the datasheet there is no specification with regards to this fact, but a damping factor ($b_m$) will be introduced in the motor’s side.
The first subset will be just composed of the motor itself. The rotor inertia \( J_R \) must be taken into consideration, to find out the torque that is given by the motor shaft (\( T_g \) represents the output torque of the motor, and the subindex 1 represents the values before reduction):

\[
T_m - b_m \dot{\varphi}_1 = T_g + J_R \ddot{\varphi}_1 \quad (6.8)
\]

The second subset will be the gearbox. This element has a reduction \( r \) and an inertia \( J_g \) (measured on the input shaft), such that the equations for this element are the following (the inertia given is the one seen by the motor [38]):

\[
T_g = T'_g + J_g \ddot{\varphi}_1 \quad (6.9)
\]

\[
T_L = rT'_g \quad (6.10)
\]

The third and last subset is the load. In this case, the load is composed by the solar panels and the whole system that supports it. The transmission is once again considered rigid. The total equivalent inertia \( J_L \) will be obtained from the CAD model. Furthermore, it will be considered that there is no torque applied to the load from the outside.

\[
T_L = J_L \ddot{\varphi}_2 \quad (6.11)
\]

\[
\varphi_2 = \frac{\varphi_1}{r}, \quad r > 1 \quad (6.12)
\]

All the previous equations can be combined in order to obtain a differential equation for the angular position of motor’s shaft (or of the solar panels, just applying the \( r \) factor):

\[
T_m - b_m \dot{\varphi}_1 = J_R \dot{\varphi}_1 + J_g \ddot{\varphi}_1 + \frac{J_L \ddot{\varphi}_2}{r}
\]

\[
\Rightarrow T_m - b_m \dot{\varphi}_1 = J_R \ddot{\varphi}_1 + J_g \ddot{\varphi}_1 + \frac{J_L \ddot{\varphi}_2}{r^2}
\]

\[
\Rightarrow \ddot{\varphi}_1 = \frac{T_m - b_m \dot{\varphi}_1}{J_R + J_g + \frac{J_L}{r^2}} \quad (6.13)
\]

The combination of both (electrical and mechanical) models will give the model of the whole motor system, including the load, so that the dynamics of the system can be expressed as mathematical equations:

\[
\frac{di}{dt} = \frac{1}{L} (U_i - K_{Emf} r \dot{\varphi}_2 - Ri) \quad (6.14)
\]

\[
\frac{d\dot{\varphi}_2}{dt} = \frac{K_m i - b_m r \dot{\varphi}_2}{J_R r + J_g r + \frac{J_L}{r}} \quad (6.15)
\]

\[
\frac{d\varphi_2}{dt} = \dot{\varphi}_2 \quad (6.16)
\]
And the energy spent to move the panel throughout one given trajectory in a period of time \(b - a\) can be expressed as:

\[
E_{\text{consumed}} = \int_{t_s}^{t_f} U_i dt
\]  

(6.17)

### 6.4 Modifications to the models

In order to be able to implement the control to the system in an appropriate manner, some of the models need to be modified.

#### 6.4.1 Generated power

The equation given in [36] (Figure 6.11, color blue) is only valid for angles between ±45° and therefore must be modified so that it goes smoothly to 0% efficiency for angles bigger than ±90° and stays at that value. In order to do this, the equation can be multiplied by \(\sigma\) (Equation 6.18), obtaining the function shown in Figure 6.11 (where the new function is represented in red).

\[
\sigma_{x,m}(\phi_{\text{incid.}}) = \frac{1}{1 + e^{\left(\frac{\phi_{\text{incid.}} - x}{m}\right)}} - \frac{1}{1 + e^{\left(\frac{\phi_{\text{incid.}} + x}{m}\right)}}
\]  

(6.18)

For this case, \(x = 63\) and \(m = 4\).

This new function solves the problem with the limited angle of incidence that the previous function had (the previous function grew to infinite for values outside the range). Nevertheless, it has some other problems regarding its shape within the ±45° range. These problems are caused by the shape of the polynomial. It has several local maxima within the range, which is not good for the optimization, nor matches the real application (the bigger the angle of incidence, the lower the efficiency). Moreover, none of the local maxima is for 0° and the function is not symmetric with respect to the vertical axis. Therefore, the polynomial has been approximated by a fourth grade polynomial (with just one maximum value, located at 0° and symmetric) multiplied by a \(\sigma\) function, and combined with a parabola to avoid local maxima in the transition to 0. This way, a new function for the efficiency (Equation 6.19) is obtained, also shown in Figure 6.11 in black color.

\[
D = f(\phi_{\text{incid.}}) = \left(\frac{\phi_{\text{incid.}}^4}{1000000} + 99.37323\right)\sigma_{67.10}(\phi_{\text{incid.}}) + \\
+ \left(1 - \left(\frac{\phi_{\text{incid.}}}{100}\right)^2\right)(1 - \sigma_{60.5}(\phi_{\text{incid.}}))
\]  

(6.19)
6.4.2 Motor model

It is very common for these kind of models to use a simplification, not taking into account the inductance of the motor when it is very small. A very small inductance causes a very fast pole in the system. This pole therefore does not have a big influence in the dynamics of the system. When using discrete models, the sampling frequency needs to be bigger than at least two times the frequency of the fastest pole. In the case of the motor that is being used in this case, the fastest pole is located at \( s = -3.25 \cdot 10^4 \) (the other two poles are located at \( s = 0 \) and \( s = -5.26 \)). Therefore the sampling frequency must be very high.

Very small sampling times may cause problems for the solvers, but also make the computation extremely slow, which would not work for real time systems. Therefore, the system can be simplified to:

\[
\frac{d\dot{\varphi}_2}{dt} = \frac{K_m i - b_m r \dot{\varphi}_2}{J_R r + J_g r + J_L r} \tag{6.20}
\]

\[
\frac{d\varphi_2}{dt} = \dot{\varphi}_2 \tag{6.21}
\]

where \( i = \frac{1}{R} \left( U_i - K_{Emf} r \dot{\varphi}_2 \right) \).

In order to check that the simplification is correct, the poles for the new system have been found, and they are located at the same places as the slower poles for the previous system. Therefore, the simplification will not have a big impact on the dynamics of the model and it is considered to be valid.
6.5 MPC formulation

Once the model has been obtained, the control problem must be formulated according to the desired behavior of the system and its constraints.

As mentioned in previous sections, the main objective of the controller is find an appropriate trade-off between the spent and the obtained energy while complying with the constraints. Therefore, the MPC formulation of the given problem is the following:

\[
J = \max_u \sum_{k=0}^{N} E_{\text{prod}}\{k\} - E_{\text{cons}}\{k\} 
\]  
(6.22)

Subject to

\[
E_{\text{prod}}\{k+1\} = E_{\text{prod}}\{k\} + t_s P_{\text{max}} f(\varphi_2\{k\} - \varphi_{\text{Sun}}\{k\}) \quad (6.23)
\]

\[
E_{\text{cons}}\{k+1\} = E_{\text{cons}}\{k\} + g(u\{k\}, i\{k\}) \quad (6.24)
\]

\[
- i_{\text{max}} \leq i\{k\} \leq i_{\text{max}} \quad (6.25)
\]

\[
- v_{\text{max}} \leq u\{k\} \leq v_{\text{max}} \quad (6.26)
\]

\[
- w_{\text{max}} \leq w\{k\} \leq w_{\text{max}} \quad (6.27)
\]

\[
E_{\text{prod}}\{0\} = E_{\text{prod},0} \quad (6.28)
\]

\[
E_{\text{cons}}\{0\} = 0 \quad (6.29)
\]

\[
\varphi\{0\} = \varphi_0 \quad (6.30)
\]

\[
\omega\{0\} = 0 \quad (6.31)
\]

Where \( f \) is Equation 6.19 and \( g \) is obtained from Equation 6.17 and the model of the motor (Equations 6.20 and 6.21) as shown below:

\[
g(u\{k\}, i\{k\}) = t_s u\{k\} i\{k\}
\]

\[
i\{k\} = \frac{1}{R} (U_i - K_{\text{Emf}} r \omega_2\{k\})
\]

\[
\omega_2\{k\} = \frac{\varphi_2\{k+1\} - \varphi_2\{k\}}{t_s}
\]

52
Chapter 7

Experiments

In this section, the implementation of the previously obtained controller algorithm will be done. This is shown in the first subsection, followed by the simulations and improvements done.

7.1 Tools

In order to check that both the models and the algorithm work as expected, some simulations have been done, using mainly two different toolboxes for MatLab: ACADO and YALMIP. The simulations have been done for both toolboxes in order to be able to chose the one that works better for the application.

Acado

The ACADO Toolkit is implemented as a self-contained C++ code, and is able to also provide a MatLab interface. It is able to solve MPC problems, also for nonlinear systems, as well as other control problems.

In order for the toolkit to be able to find the solution to the problem, six steps must be done:

1. Introduce all the variables and constants:
   For this project, and using the simplified model of the motor, there will be four DifferentialState variables (angular speed and position, as well as consumed and produced energy) and one AlgebraicState (corresponding to the current), along with one Control variable and the time declaration.
   All the parameters for the motor and transmission must be introduced as well, as normal MatLab constants.
2. Set the model:

Write the equations that rule the system. For the given application, these equations are the ones obtained for the motor, as well as for the energies (both spent and obtained). There must be four equations, corresponding to each of the \texttt{DifferentialState} variables, plus one more equation for the current.

3. Set up the objective equation:

The objective function shown in the previous section should be implemented as a \textit{Mayer term}, and the prediction horizon should be implemented as function of the time. The controller will solve an Optimal Control Problem (OCP) in each iteration of the MPC, being the length of the horizon a constant and shifting one timestep the horizon for each iteration.

4. Set up the system constraints:

There will be two main types of constraints, plus the one related to using the given model, which has to be explicitly specified. The constraints will be regarding the initial state of the system in each iteration of the MPC, and the constraints of the system throughout the control. These last constraints are the ones expressed in the formulation of the model in the previous section.

5. Create the optimization algorithm:

The created optimization algorithm has to be created and linked to the built OCP. All the custom options for the algorithm, such as the integrator used or the tolerances must be also specified here.

6. Run the code:

After compiling the previous code, apart form the C++ files that the program creates, there is also a script created that allows the user to run the code from \texttt{MatLab}'s interface. By running that script, the outputs to a specific optimal control problem are created.

Note that what is implemented is just an OCP. No feedback is implemented for this code. The way of introducing feedback is solving the OCP in a loop, changing the initial constraints at every iteration (setting them to the current real state), as well as shifting the time horizon.

Given that the code compiles and solves a specific problem, for each iteration a new compilation must be done. \texttt{ACADO} has some functions to export optimization problems so that there is no need of compiling at every iteration. Nevertheless, these functions are not available for the needed objective function in \texttt{MatLab}'s interface.

\textbf{Yalmip}

\texttt{YALMIP} \cite{yalmip} is a toolbox for \texttt{MatLab} suitable to solve optimization problems, among others. Unlike \texttt{ACADO}, it is not shipped with a low-level solver, and
therefore an external one must be installed and linked to MatLab.

In order to build and solve the problem, the following steps must be done:

1. Declare all the parameters:
   In this case, just the constant parameters of the motor, transmission and load need to be declared. As in Acado, these are expressed as normal MatLab constants.

2. Build the model:
   The model must be linear and in discrete time. In this case, the model of the motor is linear, but the model of the energies is not. Therefore, when setting the model of the motor, the energies will not be included, and they will be introduced in later steps. The model needs to be declared in state space representation.

3. Declare decision variables and initialize the controller:
   The decision variables will be the inputs (the position of the Sun for the next N timesteps) and the control signal, as well as the initial states at each timestep. The initial states are declared in this way so that they can be changed for every iteration, without building the controller every time the initial state changes. By default, Yalmip tries to minimize an objective. The initial value of this objective will be set to 0, as there will be no consumed nor produced energy right after the start.
   
   The constraints are declared by a vector. In this case, the constraints will be added to the vector at each MPC iteration.

4. Create the OCP:
   The OCP is coded as a loop in which each state is computed, and the constraints are added for each timestep. The value of the objective function is also recomputed for each iteration of the loop.

5. Build the controller:
   The last step is building the controller so that it can be called repeatedly.

Note that the code, as in Acado, builds an OCP without feedback. The controller must therefore be called in a loop, shifting the horizon and changing the initial values of each state for each timestep. Nevertheless, unlike Acado, the controller does not need to be built at every iteration, making the control algorithm faster.

Choice of the toolbox

After building the model in both of the toolboxes, some simulations have been done. According to the results obtained and the real application aspects, the decision of which toolbox to use has been made:
• ACADO’s discretization of the system is more accurate than the one \texttt{c2d}
  command does, obtaining results that are closer to the real behavior of a
  motor.

• Even if with \texttt{YALMIP} it is possible to save time by building the controller,
  the solvers tested do not handle in a fast manner the exponential function
  implemented for the efficiency calculation. Therefore, all the time that
  is saved by not needing to compile at every iteration is wasted trying
  to solve the problem with the exponential. In fact, in most cases, the
  time spent to handle the exponential is longer than the time that the
  compilation of the code in ACADO takes.

• The aim is to implement the algorithm in the Rover. The Rover is con-
  trolled by ROS, which is compatible with C++ code, but not \texttt{MatLab}
  code. ACADO compiles the \texttt{MatLab} code into C++ code, which could be
  implemented directly on the vehicle’s code. Besides, it is possible to use
  ACADO directly in C++ (and the \texttt{MatLab} interface is just a tool), while it
  is not possible to use \texttt{YALMIP} in C++.

Therefore, \texttt{Acado Toolkit} has been selected as the appropriate tool to use for
the project.

7.2 Scenarios and simulations

Once the basic model has been coded and checked, the system will be sim-
ulated in different scenarios. The aim will be testing the model in different sit-
uations, progressively introducing complexity in order to make the tests closer
to the real application.

7.2.1 Initial simulations

For the first set of simulations, the models used are the following:

• Sun’s trajectory: two different options have been used for this section.
  This is shown in the results as "phiSun". The two options are static
  (neither the Sun nor the vehicle move) and variable (either the Sun, the
  vehicle, or both are moving).

• Efficiency model: for all the simulations the efficiency is computed using
  Equation 6.19.

• Motor model: for both prediction and simulation, the simplified model
  has been used. This corresponds to Equations 6.5, 6.17, 6.20 and 6.21.

\footnotetext{1}{This is a specific problem from ACADO in this particular algorithm. Note that, with other
  solvers, it is not necessary to compile at every iteration for this problem, and for other problems
  it is not necessary to compile at every iteration with ACADO.}
For the initialization of the optimization in each iteration, only the four variables treated as differential states ($\varphi$, $\omega$, $E_{\text{cons}}$ and $E_{\text{prod}}$) are assigned a value. As the current is not treated as differential state, it is not initialized in each iteration, but just gets the value calculated for the current iteration. This case will be adjusted in upcoming sections.

**Static trajectory**

The first case to be tested will be the case in which the vehicle is not moving, and the Sun is also fixed. The reference trajectory of the Sun with respect to the Rover will therefore be a constant, which will be set with an offset. This will be done in order to check that the system is able to correct possible offsets and align itself with the optimal angle.

Note that, according to Figure 6.11, the efficiency does not have big variations when the angle of incidence is within $\pm 25^\circ$. Therefore the offset has been set high (110°), so that there is more tendency to correct the offset in order to obtain more energy.

The results obtained running this first simulation are shown in Figure 7.1. The simulation has been stopped when the angular speed is lower than a threshold (in this case, set to 0.01 rad/s, which corresponds to 39 iterations of the MPC algorithm for a prediction horizon of 15 samples).

As shown in Figure 7.1, the controller is able to correct the offset in the orientation smoothly and respecting the constraints on the states. In this case, the only constraint that has been reached is the one for the current, which saturates for the first 0.7s approximately at a value of $i_{\text{max}} = 0.72 A$. 

---

57
Variable trajectory

This simulation is valid for the following three cases:

- Sun moving and static vehicle
- Sun static and vehicle moving
- Sun moving and vehicle moving

This happens because the optimization problem is coded in the way that it just takes a trajectory from the \texttt{ComputePhySun} function, which is just a trajectory of the Sun with respect to the Rover.

For the simulation, the first trajectory selected has been a straight line with a slope of 2 rad/s. This has been selected because it is easy to make sure that the system would be able to follow it perfectly taking into consideration its constraints.

The results of this new experiment are shown in Figure 7.2. The simulation stopped when the variation of the angular speed in the last 10 timesteps is smaller than 0.01 rad/s. For this system, this happens after 42 MPC iterations.

For illustrative purposes, a slight modification of the model has been simulated. In this new simulation, the maximum power that the solar panels can

Note that, given the low speed of the Sun with respect to the Moon, this case would be very similar to the static trajectory case. Nevertheless, it will be treated in this point.
give is around 15 times smaller than in the previous ones. The results are shown in Figure 7.3. As expected, there is a constant offset between the Sun’s position and the trajectory followed by the panels. This happens because the optimization has as result that it is not worth it to spend so much energy in following accurately the trajectory for the little extra power that is going to be obtained. In this case, the length used for the simulations is the same as in the previous simulation, as that makes it possible to do appropriate comparisons.

Figure 7.3: Output of the system when the maximum power is around 15 times smaller and the input is a straight line of slope 2 rad/s

For the first simulation (Figure 7.2), the current saturates in the beginning once again. Nevertheless, when the movement is smaller (second simulation, Figure 7.3), there is no saturation of the current at any time. The power consumed in the second simulation is quite lower than in the first one, as the energy obtained is also lower and therefore it would not be a good decision to spend more energy.

The last test done for this case implies a more complex reference trajectory, which already starts with an offset with respect to the solar panels. This simulation is shown in Figure 7.4.

As shown in Figure 7.4, the algorithm is able to make an appropriate tracking of the Sun when the trajectory is more complex than just an offset or a straight line. This simulation is closer to the real application, as the Rover will have a trajectory that might not be a perfect circle (which would be the equivalent to having a straight line in the angle reference), but a more complex shape.
7.2.2 Angle management

The models used for the following simulations are the same as in the previous section. Nevertheless, some modifications to the MPC iteration algorithm regarding the management of angles have been introduced.

Until this point, the algorithm was coded in a way in which the angles can increase or decrease as much as it is required. Nevertheless, the angles should be treated in a way such that $360^\circ = 0^\circ$ and so on. Previously, it was specified that the range in which the angles are treated might vary during the optimization. This is a way of making it simpler for the algorithm to understand that angles with a difference of $360^\circ$ are the same, and that there are rotation directions to reach the same angle.

The first tried approach to take into consideration this matter was using a simple if-else clause. Nevertheless, when the solver compiles the optimizer, the values of the angles are unknown, and therefore the solver does not know if it has to apply what is inside the if or what is inside the else. Therefore, a new approach had to be used.

The next approach was using two different ranges of angles, depending on the current position of the panels. This approach was used in order to never work near the "limits" of the angles. Therefore, when the angles are close to $0^\circ$, the interval used for the angles is $[-180^\circ, 180^\circ]$ and viceversa. This approach takes the assumption that the system will not rotate more than a specified threshold throughout the prediction horizon. Therefore, just by changing the initial value when a specified threshold is trespassed, the angle of the panels with respect to the Rover will always be within the limits. For the experiments, the threshold
was set to 30° away from 0° or 180°.

This adjustment of the angles solves a problem regarding the values of the angle $\varphi_{\text{Panel,Rover}}$. Nevertheless, what mainly affects in the optimization is the difference between this angle and $\varphi_{\text{Sun,Rover}}$. In order to handle the angle variation in this sense, it was necessary the use of several flags:

- At the beginning of each iteration, the value of $\varphi_{\text{Sun,Rover}}$ at the current moment is checked.
  - In case it complies with the range of angles that the system is working on at the current moment, then no extra action is taken.
  - In case it does not comply, the operations that have to be made so that it complies are saved as flags to be done when computing the whole trajectory throughout the prediction horizon.

- After the adjustment of the angle, the angle of incidence (at the current moment) is computed. Depending on the value, there is a different flag raised, which states what should be done when checking the efficiency.

The results of some test are shown in Figure 7.5 and Figure 7.6. Note that the change of range used is not visible in the plots, as all the plots are drawn for the same range of angles.

Figure 7.5: Test 1 of the angle management: check that after 180° the angle is reset to $-180°$.

In the first test, the aim was to check that the angles were correctly "reset" during the control horizon. For the second test, the aim was checking that the algorithm was able to understand that the difference between two angles could be smaller in the opposite direction. Therefore, even if the reference is at a lower level than the initial position, the system will move the panels to a higher angle value, which will, eventually, turn negative.
7.2.3 Stop moving flag implementation

As in the previous section, this simulation only introduces changes in the code for each MPC iteration, and the models used are the same as in 7.2.1.

This implementation covers the cases in which it is not worth it to move the panels for an external reason. This reason can either be that the batteries are fully charged (and the vehicle is not spending energy in any other system, which is very unlikely) or that the vehicle is in a dark area, for example.

For both cases (and other possible scenarios), there will be a flag that will avoid calling the controller’s function, and therefore, will avoid moving the panels.

Since there is no other system implemented (external to the solar panel system), the flag will be triggered at specific timesteps in the simulations. The results of the simulation are shown in Figure 7.7.

The simulation tests the case in which the vehicle enters a dark zone in \( t = 0.7 \) s and goes to a lighted area in \( t = 2.5 \) s. As seen, when it enters a dark zone the input signal is set to 0 V, which causes the rest of the signals to progressively get their steady state values. When the flag is reset, the controller tries to follow once again the reference signal. Note that the value of the efficiency is recomputed even when the controller is off, and the generated energy is still growing due to an appropriate angle of incidence.
Parallel trajectory

A problem was encountered when doing the previous tests: with some reference trajectories, and when the angle of incidence is close to ±140°, the controller is not able to track the reference, but it moves in a parallel trajectory instead, with a shift that causes the system not to obtain energy, but just spend. An example of this bug is shown in Figure 7.8, where the system was externally stopped from the beginning and until $t = 1.4 s$.

Figure 7.7: Test of stopping flag (stops at $t = 0.7 s$ and continues at $t = 2.5 s$)

Figure 7.8: Parallel tracking problem
In order to solve this issue, several options are proposed and tested:

- Stop the system for some timesteps, in the hope that the angle of incidence changes and the controller is able to give a good solution again.
- Use a bigger prediction horizon for the controller.
- Increase the prediction horizon only when the issue is encountered, and reset it to the lower value when the performance is adequate.

In order to solve this problem, the first solution implemented is based in the use of the stop flag implemented in the previous simulations. When the system recognizes that the efficiency is below a threshold, and that it has not varied much in the last $x$ steps (where $x$ needs to be tuned), the system triggers the flag to stop the system for a few timesteps. This way, the angle of incidence varies and lets the system be able to control appropriately again. For the problem shown in Figure 7.8, the solution using this method is shown in Figure 7.9.

As shown in the simulations, the system stops the controller at about $t = 0.9s$ and turns it on again at $t = 1.4s$ following the same logic as in the previous example. Nevertheless, when it turns it on again, the system still has the same offset as before stopping. Therefore, the previous logic does not completely fix
Figure 7.10: Test with a constant reference. The first solution is implemented, and the system stops at $t = 0.9\, \text{s}$ and turns on again at $t = 1.4\, \text{s}$.

The next solution studied is the use of a larger length of the prediction horizon. This makes the optimization problem slower, but also solves (if the correct length is used) the problem. Given that very long prediction horizons would make the solver too slow, it is not desirable to have them if there is another solution to the problem. A simulation using this solution has been done and is shown in Figure 7.11.

In order not to make the controller too computationally heavy, a third solution combination of the previous two is implemented. This solution consists in increasing the horizon length only when it is necessary, using the same checking system as in the first proposed solution.

From the three proposed solutions, the third one looks at a first glance as the most appropriate, as it does not imply a waste of time. Nevertheless, with some trajectories, the problem is not solved. This is mainly due to the constraints of the system. In order to handle these cases, the solution implemented is the third one, having as backup the first one. In case that increasing temporarily the prediction horizon does not work, the system will be stopped for some timesteps, and then switched on again.

In order to be able to test the correct implementation of this option, the initial prediction horizon has been set to a low value (5 timesteps), and increased 5 timesteps each time the threshold for the efficiency was not exceeded for the
last $x$ timesteps, up to a value of 20 timesteps. In case this value is exceeded, the stopping strategy is applied. For the simulation, the changes of the horizon length according to the conditions applied are shown in Table 7.1, and the graphic result is shown in Figure 7.12.

<table>
<thead>
<tr>
<th>Time interval (s)</th>
<th>Horizon length (timesteps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>5</td>
</tr>
<tr>
<td>0.85 to 1.35</td>
<td>10</td>
</tr>
<tr>
<td>1.35 to 1.85</td>
<td>15</td>
</tr>
<tr>
<td>1.85 to 2.35</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 7.1: Changes of the length of the prediction horizon

The results show that the first three adjustments of the prediction horizon length were enough to drive the system to its desired orientation. In case a fourth one had been needed, since the backup prediction horizon would be 25, the strategy applied would have been stopping the system for some timesteps, as shown in the first proposed solution.

Note that, with a constant reference, the stopping strategy is not useful. Nevertheless, for constant trajectories, increasing the prediction horizon should be enough, and the stopping strategy is used for other trajectories for which increasing the prediction horizon might not work.

---

3 Repeating the efficiency function with $\pm 360^\circ$ offset also helps solving this issue, for some cases. This is also implemented.
Figure 7.12: Changing prediction horizon strategy. The stopping strategy was not used, as the solution was found before the timestep limit was exceeded

### 7.2.4 Current initialization

As mentioned in 7.2.1, the value of the current was not initialized for each timestep, but just computed as a result of applying a specific input for the next timestep. In this subsection, the issues related to this choice will be examined.

- The sampling time used for the OCP is too big for the simulations. Therefore, the simulation gives values of current that are not correct. When giving these values as initial states to the OCP, the solution for the problem is not correct, and can even produce unfeasibility problems. The solution for this issue would clearly be reducing the sampling time. Nevertheless, too small sampling times create a problem for the OCP, who is not able to control the system properly.

The overall solution for this problem would be using different sampling times for the simulation and for the optimization, and simulating N times for each MPC iteration (with $N = t_{s,\text{optimization}}/t_{s,\text{simulation}}$) and assigning the last values to the initialization of the next OCP.

- The model that is being simulated is not the complete model, but the simplified. This was implemented this way because of possible numerical problems with fast poles. Nevertheless, as the sampling time for the simulations will be smaller than the OCPs, this can be modified.

For the next experiments, the simulated model will be the complete one (Equations 6.5 and 6.14 to 6.17) instead of the simplified one (Equations 6.5, 6.17, 6.20 and 6.21). This will be implemented using the solution for the previous issue, as it is the only way of simulating the complete model without numerical problems.
Implementing the two previous solutions, it is possible to give an initial value for the current to the OCP. In order to check the implementation, a simulation with a fixed reference trajectory (as shown in Figure 7.1) will be done. The results of this simulation are shown in Figure 7.13.

![Figure 7.13: Implemented current initialization (simulation with complete model)](image)

The sampling time used for the plots is the same as the sampling time of the simulation (much smaller than the sampling time of the optimization), and therefore the transient of the current (and therefore of the power too) can be seen. Nevertheless, only the value every \( t_{s,\text{optimization}} \) is used as initial value for the optimization. Nevertheless, for the rest of the plots, the sampling time of the optimization will be used for the plots.

Note that in this case, the maximum allowed current is exceeded in the transient state for each timestep. This happens because the model used for the MPC is a simplified model which does not take into consideration the transients of the current. Nevertheless, this is not a problem for the real system, since the maximum value of current stated is in continuous operation and it can be surpassed for short periods of time. In this case, the maximum value it reaches is 0.79A, which is not too high with respect to the limit (0.72A).

### 7.2.5 Disturbance

This is the last simulation that has been done for testing the implementation. The model used is the same as in the previous section, i.e. the simplified model.

---

4The system has been simulated using both (complete and simplified) models. Nevertheless, given that there was no visible difference between the two simulations, only the plots where the complete model was used are shown.

5The stall current has a value of 2.15A, which is much higher than the reached current.
CHAPTER 7. EXPERIMENTS

(Equations 6.5, 6.17, 6.20 and 6.21) for the optimization and the complete model (Equations 6.5 and 6.14 to 6.17) for the simulation.

In this simulation, a disturbance is introduced at a specific timestep. The disturbance will consist in a sudden change of the orientation of the panels. It will just affect the system in one specific timestep and the aim is to check that the controller is able to adapt to the new state and correct the position to follow the trajectory again. The results of the simulation are shown in Figure 7.14.

![Figure 7.14: Evolution when a disturbance is applied. Disturbance of 180° offset at t = 2s](image)

For the simulation, a disturbance was introduced at \( t = 2s \), with a value of 180° offset from the position in which it was originally. As shown, the system is able to correct the position and drive back the panels to their desired orientation.

### 7.2.6 Complete simulation

In the previous parts, each case was presented, simulated, and solved (in case it was necessary) separately. Now a long simulation using the simplified model (Equations 6.5, 6.17, 6.20 and 6.21) for the optimization and the complete model (Equations 6.5 and 6.14 to 6.17) for the simulation is done. In this case, disturbances (7.2.5) and stopping (7.2.3) have been introduced at different timesteps. The correction of the parallel tracking (7.2.3, Parallel tracking) was also implemented, in case it was required at some steps. Finally, the reference trajectory used is the composition of the movement of the Sun and the movement of the Rover.

Given that the real movement of the Sun with respect to the Moon is too slow to be seen in a simulation (its period is of 29.5 days), it has been accelerated to a period of 80s so that it can be appreciated in the composition of the simulation.
The movement of the vehicle in the surface of the Moon has been set to an oscillatory trajectory, in such way that the angle of the Rover with respect to the Moon follows the equation \( \phi_{Rover,\text{Moon}} = 1.2 \sin(1.5t) \). The simulation ran for 150s, and during that time, two dark areas and a disturbance were simulated. The results are shown in Figure 7.15.

After analyzing the results, it was seen that the correction of the parallel tracking was not used. It might happen that with other stopping times, disturbance time or magnitude or trajectory of the Rover, the correction is needed, and therefore it will be left in the code, as a back-up.

Since it is difficult to appreciate the different cases given the long time the simulation is run for, a detail (only on \( \phi \)) for each factor is shown in Figures 7.16 to 7.18.

For both dark areas, the system is able to quickly get back to track the reference angle. It is interesting to observe the different effect on the efficiency that the length of the dark areas cause. In the case of the first dark area Figure 7.16, the estimation of the efficiency starts varying between 0 and 100%, as it happens in the second one Figure 7.17. Nevertheless, the first...
one is long enough to suffer the effect of the Sun’s movement with respect to the Moon, and at some point it starts oscillating between 0 and other values lower than 100%. The second is always close enough to the reference trajectory to obtain 100% efficiency in the points where the sine signal gets close to the orientation of the panels.

Note that, even if the panel is supposed to be in a dark area, the algorithm is assuming that it can still get energy. This happens because the only thing that the stopping flag does is shut down the controller. The rest of the algorithm has no knowledge on the status, and therefore thinks that there is still sunlight. This fact does not have any influence on the well-functioning of the whole system, since the energy obtained up to the current moment will not influence the decisions of the MPC. The MPC will still try to maximize the obtained energy with respect to the consumed energy, regardless of the energy balance at each moment.

For the disturbance (Figure 7.18), the controller is also able to quickly drive the panels to track the reference. Note that, as the disturbance introduced is big, the efficiency drops suddenly to 0, and then it slowly gets to 100%, while the angle gets closer to the trajectory.
Chapter 8

Results and Conclusions

To finish with the control part, some final results will be shown. In order to test the trajectory proposed in Figure 6.4, the simplified model (Equations 6.5, 6.17, 6.20, and 6.21) for the optimization and the complete model (Equations 6.5 and 6.14 to 6.17) for the simulation are used.

In previous sections, the trajectory of the Sun with respect to the Rover was manually introduced so that the tests could be easily done. Now a simulation introducing just a trajectory for the Rover will be done, using the most complete version of the algorithm (using all the improvements implemented separately in the previous section).

The first step for the algorithm will be now find the angle of the Rover with respect to the Rover as a function of time. In order to do so, the speed of the Rover has been set to 0.2m/s, and the tangent to the trajectory (and therefore the angle of the Rover with respect to the Moon’s coordinate system) has been found for the position of the Rover every 20s. This value has experimentally proven to be suitable for this speed of the Rover, having a smooth shape without taking more computational time than it is required. For the trajectory shown in Figure 6.4 and a speed of 0.2m/s, the evolution of the angle with respect to time is shown in Figure 8.1.

![Figure 8.1: Angle of the Rover with respect to time for the previously given trajectory](image)

72
Given that the algorithm needs a continuous function, and that a very complex function would make the solver take an unnecessarily long time, the values of the angles have been approximated as a two degree polynomial. This polynomial is re-built every 20s as follows:

1. Initially, the first three values of the angle are used to build the polynomial.
2. When the simulation reaches the middle of the interval (i.e, 20s using thee points and the time intervals used for extracting the orientations), the three used values shift one position, so that the second, third and fourth values of orientation are used.
3. This is repeated until the end of the simulation.

The simulation has run for the whole trajectory. Given the length of the whole trajectory, the necessary time for the Rover to complete it was around 700s. The angle of the Rover with respect to the Moon has been converted into the angle of the Sun with respect to the Rover, taking into consideration the position of the Sun with respect to the Moon as well, as shown in Figure 6.2 (Equation 6.3). The results of the simulation are shown in Figure 8.2.

![Figure 8.2: Simulation of the system for the previously given trajectory](image)

From this simulation, as well as from the previous ones, the following conclusions have been extracted:

- Note that, in Figure 8.2, the angle seems to be mirrored with respect to Figure 8.1. This happens because of the transformation of the coordinate system expressed in Equation 6.3, which changes the sign of $\varphi_{\text{Rover,Moon}}$, apart from combining it with the movement of the Sun with respect to the Moon, which is, for a short time interval, not noticeable.
- The Sun’s orientation is followed very accurately, when simulating. Nevertheless, the real system will have static friction and possibly other facts
that will cause the tracking less accurate. Therefore, a low level controller could be used in conjunction with the tracker in order to obtain better performance.

- Regarding the transformation of the orientation sequence into a polynomial, in the graphs it can be seen that when the polynomial is changed (following the previously explained algorithm), the controller produces a small "jump" on the inputs required (this can be seen in detail in Figure 8.3). The "jumps" are caused by the change of curvature of the polynomial, and therefore are bigger when the function turns from concave to convex, and vice versa. This issue is not affecting negatively in a big magnitude to the orientation. Nevertheless, if desired, the "jumps" could be made smaller by creating polynomials of bigger degree. Note that this would make the system slower, as it introduces complexity.

![Figure 8.3: Detail of the "jumps" produced by the change of polynomial. When the low degree polynomial for the orientation of the Rover is changed, the control input shows sudden changes (easily observable at $t = 240s$ and $t = 260s$ for this case)](image)

- For the given trajectory, the required rotational speed is very low. Therefore, the effect of the static friction in the real system will be more noticeable.

In order to check the correction of the "jumps, a simulation having a much faster vehicle (so that the simulation is quicker) and polynomials of degree 4 has been done. These results are shown in Figure 8.4. Note that, due to the higher speed of the vehicle, the values of the inputs are much higher (the angle changes faster). The sampling time for obtaining the input sequence has also changed in such way that the angle is computed for every 4 meters driven (same relation sampling-speed as in the previous simulation).

One last remarkable thing in this result is the fact that the tracking is not as perfect as it look before. This is a result of the optimization. The tracking does not follow so closely the Sun’s trajectory because it is not optimal to do so when the angular speed of the Sun with respect to the Rover is that high.
Figure 8.4: Results when the Rover moves 80 times faster and the polynomial has degree 5. The sudden changes of input are reduced thanks to the increase of the degree. The tracking is less accurate because of the speed of the Rover and the maximum rotational speed of the panels.
Part III

Results and Conclusions
Chapter 9

Results

Since the results for each part have been shown in the corresponding part, just a summary about them will be done here.

Mechanical part

In this part, the hardware of the developed system was studied and designed. The solar panel system needed to be deployable, so that it took little space when launching, but also needed a large surface in order to obtain enough energy to charge the batteries for the whole vehicle. Therefore, the state of the art in deployable systems was analyzed and a rollable mast was chosen as the solution to implement.

Along with the rollable mast, and due to the large surface required for the application, extra surface was designed to be deployed in a second phase. This surface would consist in two flaps at the laterals of the main panel, which would allow to reduce the height of the panels.

The designed panel’s support incorporated a system that allowed it to rotate around its vertical axis, providing one degree of freedom for solar tracking and energy optimization. Due to the mission’s aim, only one degree of freedom was required, simplifying the construction of the system, but allowing enough control freedom.

It was possible to fulfill all the requirements stated in section 2.3 giving special importance to the ones which would cause major problems otherwise. Some examples of this are requirements 2 (size of the mechanism), 3 (main position of the panels) and 7 (clash-free). Not fulfilling requirement 2 would not allow to mount the system on the Rover. Not fulfilling requirement 3 would not allow the system to get energy. Finally, not fulfilling requirement 7 would cause the system to collide against itself, possibly breaking parts of it.
The implementation in CAD of the fully deployed solar system in the vehicle is shown in Figure 9.1. As seen, the panels have quite a big size with respect to the vehicle, making it more complicated to fit it in a simple way and more complicated to allow it to move without clashing. Therefore, when modifying the Rover, the panel system should also be taken into consideration.
CHAPTER 9. RESULTS

Control part

At the beginning of this part, the most used solar tracking techniques were studied. Many of the techniques commonly applied are based on equations that show the movement of the Sun. This happens both in household applications and power plants, where the high-level controllers are used to maintain the variables around a setpoint, instead of focusing on the correct tracking of the Sun.

Regarding the requirements, a similar approach was taken, using MPC control to find the best orientation sequence to follow given a theoretical evolution of the Sun’s orientation (based in equations).

In order to be able to implement the controller, the models of all the variables had to be found as mathematical equations. These variables include the movement of the Rover and the movement of the Sun and the model of the motor, between others.

The controller was implemented using Acado toolkit for MatLab and tested in different scenarios. For each scenario, the result was analyzed and the necessary modifications were implemented in order to fix problems or improve the performance.

Regarding some of the requirements, specifically requirements 3 (zero-energy places) and 4 (fully charged battery), some modifications to the control algorithm, not included in the MPC formulation, had to be implemented.

Once all the found problems were fixed and all the requirements were implemented, the system was simulated for a specific trajectory, obtaining some results that showed the well-functioning of the system.
Chapter 10

Conclusions

Lastly, in this final chapter, the conclusions of the whole project will be shown. These conclusions include both parts of the project, along with some conclusions on the work environment.

The first conclusion that is worth to mention, and that includes both parts of the project, is the fact that when working with space-related topics, the restrictions and requirements of everything used are much harder and complicated (and also more) than when working with other topics. The systems that travel to space need to be more robust and fail-proof than other systems, as they cannot usually be maintained in case of failure, and the cost of sending them into space is high.

In what regards to work environment, it is important to realize that industry projects are usually much more complicated than university projects. This means that there are a lot of external factors that each person has to take into consideration, and communication and coordination between team-members is highly important. Besides these facts, it is not always easy to get everybody to agree in some aspects of the project, which is not ideal for the project.

Communication between team-members is also very important, as being able to get up-to-date information is very useful in order not to waste time and effort unnecessarily.

Regarding the project itself, some conclusions on each chapter have been developed.

For the state of the art, in the mechanical part, there are a lot of deployment methods for space applications. These methods have been being studied since the beginning of the era in which humans started to send satellites, rockets or vehicles to space, as making things compact makes the launch cheaper (and the space industry is already quite expensive). Therefore, as years have gone by, some simpler, cheaper, reliable and lighter solutions have been born. This means that more solutions will be found in the next years keeping this tendency,
as technology improves.

For the control part, it seems like simple controllers are usually chosen (or sometimes even no control). For spacial applications, control is usually implemented in order to get more energy. Nevertheless, it seems like energy harvesting in space is evolving into some different ways, such as nuclear energy instead of photovoltaic energy.

Regarding the requirements, for both parts, some simple specifications have been set. These first specifications had some impact in the desired behavior of the system even if they were simple. After these requirements were fulfilled, some extra requirements were chosen in order to correct some problems or to improve the behavior in some situations. For example, in the mechanical part, a clash-free requirement would not be necessary if the panels did not have to rotate. And in the software part, there would not be a need of a requirement for zero-energy places if no trade-off was needed.

Lastly, with regards to the development of the project, in both cases the first step has been to create a base idea. This base idea was designed and implemented, and later modified in order to cover the necessities. The modification are specially noticeable in the software part, where the improvements have been implemented on top of the base algorithm.

10.1 Future work

Some ideas on the future work to be done for both parts are developed below:

For the hardware part, the next thing to do would be build a prototype of the system. Note that, there is already an existing prototype (scale 1:1 shown in Figure 10.1) of the Rover. Nevertheless, this prototype only includes some parts of the real designed solar system, having some of the parts in a simplified way. Therefore, a prototype including all the parts of the solar system should be built in order to check the deployment and software.

Once the prototype is built and checked, the system should be built for the real application. Unfortunately, there is currently no real Rover. Therefore, this task will have to be postponed until there is an available vehicle.

Regarding the software part, the robustness of the system should be checked with respect to end-time constraints and other facts. A stability analysis needs also to be done for the proposed algorithm.

The software should be checked in the prototype in order to check the validity of the simulations and tweak the variables of the system until the results are satisfactory.

Other thing that could be analyzed for the system are some different scenarios and some solutions should be found in case they are needed. Lastly, the system should be implemented in C++ and included in the code of the rest of
the vehicle. When implementing in C++, the solver should be precompiled so that there is no need of compiling it for every MPC iteration. By doing this, the solving speed should be incremented.

Note that, when simulating the system, the time was too high to be executed in real time. Nevertheless, with the real system, there is no need of simulating it at every iteration, which would reduce the execution time. Creating a precompiled solver, the system would also be faster. One iteration of the MPC (without taking into consideration compilation time) takes approximately 0.372s. This is higher than the implemented sampling time, which would not allow to use the system in real time. Nevertheless, when implementing in C++ the time will be reduced. In order to fix this issue, the proposed solutions are increase the sampling time and simplifying the code (which would probably mean losing accuracy for the tracking).
Bibliography


List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>XSAS scissor deployment model</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Deployed FalconSat-7</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Pantograph mechanism</td>
<td>8</td>
</tr>
<tr>
<td>2.4</td>
<td>Truss structures</td>
<td>9</td>
</tr>
<tr>
<td>2.5</td>
<td>InflateSail project</td>
<td>9</td>
</tr>
<tr>
<td>2.6</td>
<td>QuakeSat in deployed configuration</td>
<td>10</td>
</tr>
<tr>
<td>2.7</td>
<td>Skew fold configurations</td>
<td>11</td>
</tr>
<tr>
<td>2.8</td>
<td>Wooden model of solar origami deployment system</td>
<td>11</td>
</tr>
<tr>
<td>2.9</td>
<td>Different configurations of tape spring booms</td>
<td>12</td>
</tr>
<tr>
<td>2.10</td>
<td>Synchronous system for deployment with tape spring composite</td>
<td>12</td>
</tr>
<tr>
<td>2.11</td>
<td>Antenna deployment with composite boom</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Base plate of the solar deployment system</td>
<td>18</td>
</tr>
<tr>
<td>3.2</td>
<td>Simulation of the base plate</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>Shaft</td>
<td>19</td>
</tr>
<tr>
<td>3.4</td>
<td>Coupling of the shaft to the base plate</td>
<td>20</td>
</tr>
<tr>
<td>3.5</td>
<td>Connection of the bearing case to the base plate</td>
<td>20</td>
</tr>
<tr>
<td>3.6</td>
<td>Motor and plate coupling</td>
<td>21</td>
</tr>
<tr>
<td>3.7</td>
<td>Motor and plate coupling</td>
<td>21</td>
</tr>
<tr>
<td>3.8</td>
<td>Flexible carbon fiber sheet used as support for the panels</td>
<td>23</td>
</tr>
<tr>
<td>3.9</td>
<td>Motor and coupling system</td>
<td>24</td>
</tr>
<tr>
<td>3.10</td>
<td>Upper holder for the panels</td>
<td>25</td>
</tr>
<tr>
<td>3.11</td>
<td>Lateral deployment phase</td>
<td>26</td>
</tr>
<tr>
<td>3.12</td>
<td>Boom deployment system</td>
<td>27</td>
</tr>
<tr>
<td>4.1</td>
<td>Deployment sequence of the solar panel system</td>
<td>29</td>
</tr>
<tr>
<td>5.1</td>
<td>Scheme of the tracking strategies implemented for simple systems</td>
<td>34</td>
</tr>
<tr>
<td>5.2</td>
<td>Dual axis tracking</td>
<td>35</td>
</tr>
<tr>
<td>5.3</td>
<td>Orbiting satellites with solar panels</td>
<td>37</td>
</tr>
<tr>
<td>5.4</td>
<td>Spacial vehicles with horizontal solar panels</td>
<td>37</td>
</tr>
<tr>
<td>6.1</td>
<td>Overview of the controller’s scheme</td>
<td>41</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2</td>
<td>Computation of the orientation of the rover with respect to the Sun.</td>
</tr>
<tr>
<td>6.3</td>
<td>Trajectory of the Sun w.r.t the South pole of the Moon for one cycle.</td>
</tr>
<tr>
<td>6.4</td>
<td>Example of desired trajectory for the Rover.</td>
</tr>
<tr>
<td>6.5</td>
<td>Change of coordinates scheme.</td>
</tr>
<tr>
<td>6.6</td>
<td>Computation of the energy obtained from a certain input sequence.</td>
</tr>
<tr>
<td>6.7</td>
<td>Impact of the angle of incidence on the efficiency of the panels.</td>
</tr>
<tr>
<td>6.8</td>
<td>Computation of the energy consumed when having a specific sequence of inputs.</td>
</tr>
<tr>
<td>6.9</td>
<td>Electrical model of the motor.</td>
</tr>
<tr>
<td>6.10</td>
<td>Mechanical scheme of the motor.</td>
</tr>
<tr>
<td>6.11</td>
<td>Impact of the angle of incidence on the efficiency of the panels valid for all angles.</td>
</tr>
<tr>
<td>7.1</td>
<td>Output of the system when there is an offset of value 110° in the orientation.</td>
</tr>
<tr>
<td>7.2</td>
<td>Output of the system when there is a straight trajectory (speed: 2 rad/s).</td>
</tr>
<tr>
<td>7.3</td>
<td>Output of the system when the maximum power is around 15 times smaller and the input is a straight line of slope 2 rad/s.</td>
</tr>
<tr>
<td>7.4</td>
<td>Output of the system when using a more complex reference trajectory.</td>
</tr>
<tr>
<td>7.5</td>
<td>Test 1 of the angle management: check that after 180° the angle is reset to −180°.</td>
</tr>
<tr>
<td>7.6</td>
<td>Test 2 of the angle management: check that it considers both turning directions and takes the smaller.</td>
</tr>
<tr>
<td>7.7</td>
<td>Test of stopping flag (stops at t = 0.7s and continues at t = 2.5s).</td>
</tr>
<tr>
<td>7.8</td>
<td>Parallel tracking problem.</td>
</tr>
<tr>
<td>7.9</td>
<td>Fixed parallel tracking with the first solution.</td>
</tr>
<tr>
<td>7.10</td>
<td>Test with a constant reference.</td>
</tr>
<tr>
<td>7.11</td>
<td>Test with a constant reference and larger prediction horizon.</td>
</tr>
<tr>
<td>7.12</td>
<td>Changing prediction horizon strategy.</td>
</tr>
<tr>
<td>7.13</td>
<td>Implemented current initialization (simulation with complete model).</td>
</tr>
<tr>
<td>7.14</td>
<td>Evolution when a disturbance is applied. Disturbance of 180° at t = 2s.</td>
</tr>
<tr>
<td>7.15</td>
<td>Results for a simulation of 150s (3000 timesteps).</td>
</tr>
<tr>
<td>7.16</td>
<td>First dark area: from t = 10s to t = 50s.</td>
</tr>
<tr>
<td>7.17</td>
<td>Second dark area: from t = 60s to t = 70s.</td>
</tr>
<tr>
<td>7.18</td>
<td>Disturbance at t = 110s, putting the system 180° away from its orientation.</td>
</tr>
<tr>
<td>8.1</td>
<td>Angle of the Rover with respect to time for the previously given trajectory.</td>
</tr>
<tr>
<td>8.2</td>
<td>Simulation of the system for the previously given trajectory.</td>
</tr>
</tbody>
</table>
8.3 Detail of the "jumps" produced by the change of polynomial . . . 74
8.4 Results when the Rover moves 80 times faster and the polynomial has degree 5 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
9.1 Assembly of the panel system in the vehicle . . . . . . . . . . . . 78
10.1 Prototype of the Rover . . . . . . . . . . . . . . . . . . . . . . . 82
List of Tables

2.1 Summary of the studied deployment methods . . . . . . . . . . . 14
3.1 Characteristics of the base plate . . . . . . . . . . . . . . . . . 19
3.2 Characteristics of the shaft . . . . . . . . . . . . . . . . . . . . 20
3.3 Characteristics of the outer bearing case . . . . . . . . . . . . . 22
3.4 Characteristics of the top inner bearing case . . . . . . . . . . . 22
3.5 Characteristics of the bottom inner bearing case . . . . . . . . . 22
3.6 Characteristics of the motor holder . . . . . . . . . . . . . . . . . 22
3.7 Characteristics of the cells’ support . . . . . . . . . . . . . . . . . 23
3.8 Pros and cons of each method for lateral deployment . . . . . . 24
3.9 Characteristics of the upper holder for the central panel . . . . . . 25
3.10 Characteristics of the upper holder for the lateral panels . . . . . 25
7.1 Changes of the length of the prediction horizon . . . . . . . . . . 66
Appendix A

Listings

The code used for simulating the system can be found in [https://github.com/CeciliaMartinezMartin/ThesisSolarPanel.git](https://github.com/CeciliaMartinezMartin/ThesisSolarPanel.git)

The code for the Yalmip solver is shown in this section. This code has not been used for simulations, but just for testing which solver gave better results.

```matlab
1 % System's parameters
2 yalmip('clear')
3
4 Pmax = 153; % max power (W)
5 L = 0.0857e-3; % Motor terminal inductance (H)
6 Kemf = 1/(1700*2*pi/60); % speed constant (1/(rad/(sV)))
7 R = 2.79; % Motor terminal resistance (ohm)
8 Km = 5.62/1000; % torque constant (Nm/A)
9 Jr = 1e-7; % rotor inertia (kgm2)
10 Jg = 0.161e-7; % gear inertia (kgm2)
11 Jl = 0.0387; % load inertia (kgm2)
12 n = 138; % reduction
13 imax = 0.72; % maximum continuous current (A)
14 vmax = 6; % maximum input voltage (V)
15 wmax = 6770*2*pi/60; % maximum angular speed (rad/s)
16
17 wSun = 2;
18 b = Km*Kemf/R;
19
20 % Model data (motor)
21 Ts = 0.05;
22 N = 20;
23 T = Ts*N;
24
25 A = [(-Km/((Jr*n)+(Jg*n)+(Jl/n)))*(Kemf*n/R+b*n/Km) 0;... 
26 1 0];
27 B = [(Km/((Jr*n)+(Jg*n)+(Jl/n)))/R; 0];
28 C = eye(size(A, 1));
29 D = zeros(size(B, 1), size(B,2));
30
31 sys = ss(A, B, C, D); % Build the state space model
32 dsys = c2d(sys, Ts, 'zoh'); % Discretize the model
33 Ad = dsys.A; % Get discrete system matrices
34 Bd = dsys.B;
```

90
% MPC data and decision variables
nx = size(A, 1); % Number of states
nu = size(B, 2); % Number of inputs
x0 = sdpvar(nx, 1); % Initial state at each point
ec0 = sdpvar(1, 1); % Initial value of consumed energy
ep0 = sdpvar(1, 1); % Initial value of generated energy
u = sdpvar(repmat(nu,1,N),repmat(1,1,N)); % Controls
phiSun = sdpvar(repmat(1, 1, N+1), repmat(1, 1, N+1)); % Relative position of the Sun

% Initialization of the problem
constraints = [];
objective = 0;
x{1} = x0;
econs{1} = ec0;
eprod{1} = ep0;

% Control loop
for k = 1:N % Loop through the control horizon
x{k+1} = Ad*x{k} + Bd*u{k}; % Motor equations
i{k} = (u{k}- Kemf*x{k}(1) *n)/R; % Current
p{k} = u{k}*i{k}; % Power
econs{k+1} = econs{k} + Ts*p{k}; % Consumed energy
objective = objective + Ts*p{k}; % Minimize the consumed energy
eprod{k+1} = eprod{k} + Ts*ComputeEff(x{k}(2), phiSun{k}); % Produced energy
objective = objective - Ts*ComputeEff(x{k}(2), phiSun{k}); % Maximize the produced energy

% Set the constraints
constraints = [constraints, -vmax <= u{k} <= vmax];
constraints = [constraints, -wmax <= x{k}(1) <= wmax];
constraints = [constraints, -imax <= i{k} <= imax];
end

% Build the controller
parameters_in = {x{1}, [phiSun{:}], eprod{1}, econs{1}};
parameters_out = {{u{:}}, {x{:}}, {p{:}}, {i{:}}, {econs{:}}, {eprod{:}}};
opts = sdpsettings('verbose',2, 'solver', 'fmincon');
controller = optimizer(constraints, objective, opts, parameters_in, parameters_out);

x = [0;0];
Epin = 0;
Ecin = 0;
Econsumed = [0];
cif;
xhist = x;
uhist = [1];
phist = [1];
hhist = [1];
Echist = [1];
Ephist = [1];
len = 150/Ts;
for i = 1:len
inputs = {x, ComputephiSun(Ts*(i-1):Ts:T+Ts*(i-1), wSun), Epin, Ecin};
phiSun = ComputePhisun(Ts*(i):Ts:T+Ts*(i), wSun);
[solutions, diagnostics] = controller(inputs);
U = solutions(1);
X = solutions(2);
P = solutions(3);
I = solutions(4);
Ec = solutions(5);
Ep = solutions(6);
uhist = [uhist U(1)];
phist = [phist P(1)];
ihist = [ihist I(1)];
Echist = [Echist Ecin];
Ephist = [Ephist Epin];
Epin = Epin+Ts*ComputeEff(X(2,1), phiSun(1));
Ecin = Ecin+U(1)*I(1)*Ts;
if diagnostics ==1
    error('The problem is infeasible');
end
x = Ad*x+Bd*U(1);
xhist = [xhist x];
pause(0.05)
Appendix B

Drawings

The drawings of the following pieces are shown:

- Base plate
- Base plate connection
- Outer bearing case
- Inner top bearing case
- Inner bottom bearing case
- Motor holder
- Central panel upper holder
- Lateral panel upper holder
\( \phi 52.00 \) Actually 62, needs tolerances: -14, -33

**Material:** Aluminium

**Title:** Outer bearing case

**SCALE:** 2:1

**A3**

**Weight:**
Material: Aluminium

Title: Inner bottom bearing case

Scale: 5 : 1

Dimensions:
- ø31.00
- ø20.00
- ø10.00
- 5.00
- 2.50
- 1.45
- 0.45
- 0.80 x 45°
Motor holder
Aluminium

Material: Aluminium

Dimensions:
- 25.00
- 7.00
- 6.00
- 5.00
- 4.00
- 3.00
- 2.00
- 1.00

UP 90° R 2
DOWN 90° R 2
UP 90° R 2
DOWN 90° R 2

Notes:
- Scale: 1:1
- Sheet: 1 of 1
- Weight: 66.00
- X: 578
- Y: 324
- Z: 442

Dimensions in millimeters.