

TECHNICAL UNIVERSITY OF CRETE

DIPLOMA THESIS

Moving windows value iteration for effective solar tracking.

Author:

Giorgis-Orfeas ROMAIDIS

Supervisor:

Assistant Prof. Georgios
CHALKIADAKIS

School of Electronic and Computer Engineering

Dissertation Thesis Committee

Assistant Prof. Georgios Chalkiadakis

Assistant Prof. Eftichios Koutroulis

Associate Prof. Michail Lagoudakis

July 2014



Πολυτεχνείο
Κρήτης

TECHNICAL UNIVERSITY OF CRETE

Abstract

School of Electronic and Computer Engineering

Moving windows value iteration for effective solar tracking.

by Giorgis-Orfeas ROMAIDIS

Solar trackers have recently become an important asset in energy production industry. The most efficient type is dual axis trackers that follow sun's movement in horizontal and vertical level based on an astronomical equation. In this thesis we present several variants of a dynamic programming method to calculate the optimal solar tracking strategy of a given tracker. To this end, we first make sure that we appropriately incorporate the physical characteristics of a commercially available PV system in our model. Then we choose a combination of specific dates and weather data, both historic and artificial, to use as datasets for experimentation over typical weather patterns. Our proposed algorithm addresses the problem of systems positioning as a sequential decision making problem for optimal control. We use a Markov Decision Process (MDP) representation and apply value iteration (VI) with a k-step look ahead functionality over reduced state-space windows, to approximate an optimal solution. The results show that the performance of our approach clearly outperforms that of a typical dual axis tracker.

Acknowledgements

I would like to thank Assistant Prof. Georgios Chalkiadakis for the advise, encouragement and support he provided to me in supervising this thesis. Also, I would like to thank all the members of this dissertation committee, Assistant Prof. Eftichios Koutroulis and Associate Prof. Michail Lagoudakis, for their collaboration and valuable comments. Most of all, I would like to thank my family for their continuous support.

Contents

Abstract	ii
Acknowledgements	iii
List of Figures	vi
List of Tables	vii
Abbreviations	viii
1 Introduction	1
2 Background and Related work	3
2.1 Motivation	3
2.1.1 Towards a Smart Grid	3
2.1.2 RENES: A WDERs power output prediction tool	4
2.2 Photovoltaic and solar tracking technology	5
2.2.1 Solar radiation model	5
2.2.2 Photovoltaic operational principles	6
2.2.3 PVS tracking solutions	6
Single axis tracking	7
Dual axis tracking	8
2.3 Markov Decision Process-MDP	9
Dynamic programming	11
2.4 Related Work	13
3 Methodology	18
3.1 Real world problem representation	18
3.1.1 Physical aspects of the system	18
3.1.2 Discretization of continuous space-time	19
3.1.3 MDP representation	20
3.2 Datasets	22
3.2.1 Reasoning of selected dates	22
3.2.2 Astronomical aspects of the selected dates	25
3.2.3 Dataset construction	33

3.3	Moving windows method	35
3.3.1	Design and properties	35
	Special cases for $\kappa_{sc}, \lambda_{sc}$	38
3.3.2	k -step look ahead	38
4	Simulation results	42
5	Renop- a Web User Interface	69
6	Conclusions and future work	72
Bibliography		73

List of Figures

2.1	Passive Tracker	7
2.2	North-South	8
2.3	East-West	8
2.4	Dual axis tracker	8
3.1	Cartesian-like sun route for 21 December	25
3.2	Cartesian-like sun route for 21 June	27
3.3	Cartesian-like sun route for 20 March	29
3.4	Cartesian-like sun route for 22 September	31
3.5	Available states from s_c at t	39
3.6	Available states from s'_1 at $t + 1$	39
3.7	Available states from s''_2 at $t + k$	39
4.1	k-step lookahead comparison	64
4.2	Graphical results for 21 December with complete overcast.	67
4.3	Graphical results for 21 June 2012 based on historical data.	68
4.4	Graphical results for 22 September 2012 based on historical data.	68
5.1	Renop - WindowVI option.	69
5.2	Renop - policy iteration method.	70
5.3	Renop results page.	70

List of Tables

3.1	Physical dimensions	20
3.2	Space-time characteristics	20
3.3	Sky cover	23
3.4	5-year sky cover history for 21 December	23
3.5	5-year sky cover history for 21 June	24
3.6	5-year sky cover history for 22 September	24
3.7	5-year sky cover history for 20 March	24
4.1	Results for 20 March 2011	42
4.2	Results for 22 September 2012	44
4.3	Results for 21 June 2012	45
4.4	Results for 21 December 2008	46
4.5	Results for 20 March 2011, scenario [CLR-OVC]	48
4.6	Results for 22 September 2012, scenario [OVC-CLR]	49
4.7	Results for 20 March 2011, scenario [FEW-BKN-FEW-BKN]	51
4.8	Results for 21 December 2008, scenario [FEW-BKN-FEW-BKN]	52
4.9	Results for 20 March 2011, scenario [BKN-FEW-BKN-FEW]	53
4.10	Results for 21 December 2008, scenario [BKN-FEW-BKN-FEW]	55
4.11	Results for 21 December 2008, scenario [OVC]	56
4.12	Results for 20 March 2011 with wind speed $60km/h$	57
4.13	Results for 22 September 2012 with wind speed $60km/h$	58
4.14	Results for 21 June 2012 with wind speed $60km/h$	60
4.15	Results for 21 December 2008 with wind speed $60km/h$	61
4.16	Accumulated energy production results.	63
4.17	Results for different dynamic programming approaches. Values are in KWh and correspond to net energy gain.	65

Abbreviations

AI	A rtificial I ntelligence
VI	V alue I teration
MDP	M arkov D ecision P rocess
PVS	P hoto V oltaic S ystem
WDER	W eather D riven and non-scheduled E nergy R esources
POMDP	P artially O bservable M arkov D ecision P rocess

This is dedicated to my family and my friends. . .

Chapter 1

Introduction

New energy production technologies rapidly emerged over the last decades. The traditional fossil-based sources are critically reviewed for two main reasons, environmental pollution in various forms and supply limitations, even though not immediate. As a result, the solutions developed are using nature's renewable sources, like solar radiation, winds and sea waves, and are generally called *Weather-Driven and non-scheduled Energy Resources-WDERs*-. Gradually these systems were incorporated into the traditional power grid arising certain issues. Since the fundamental operation principle of these systems relies on weather conditions it is obvious that the produced amount of energy is remittent in contrast with fossil fuel power plants. Hence, improved efficiency and energy output prediction tools would assist in addressing this issue.

This thesis focuses on photovoltaic systems and presents a method for enhancing energy output and estimating the total energy gain, by approximating the optimal solar tracking strategy for each day. The main contribution is that the real world problem is modeled in detail, for a software simulation environment of our design, and then is treated as an optimal control problem with artificial intelligence methods. Also, in contrast to other approaches, here we account for the consumption of the solar tracking system. Moreover we take into account forecasted weather conditions in order to produce the best possible daily solar tracking strategies. In particular, an MDP framework was used and *value iteration(VI)* applied for approximating nearly optimal solutions corresponding to *photovoltaic systems(PVS)* orientation through the day; thus, in contrast to others, we aim to produce tracking schedules on a daily rather than on a long-term basis. In order to improve results further a *k*-step look ahead functionality was added. This combination has never been proposed and tested before because the necessary reward model was missing, but in this thesis we make use of the RENES¹ estimator for this task. RENES

¹<http://www.intelligence.tuc.gr/renes/>

was developed in the Intelligence Lab of the School of Electronic and Computer Engineering in TUC as part of the diploma thesis of Aris Athanasios Panagopoulos. RENES exploits free-to-use online weather forecasts in order to estimate PVS energy production output. For this reason, is used here in order to feed our methods with the necessary rewards that correspond to estimated production output. This unique approach opens a new research area available for further experimentation and optimization.

In Chapter 2 the necessary theoretical background and related literature are presented. Subsequently the methodology followed is discussed in Chapter 3. The representation of the real world model is built. Physical aspects of the system are described, space and time entities are handled and then all of these are integrated into the selected mathematical framework. In order to evaluate this work specific datasets are produced based on various dates and weather conditions combinations. Finally the proposed algorithm with its components is described. The experimental results are presented and discussed in Chapter 4. In Chapter 5 we present a web user interface we developed in order to compare the efficiency of dynamic programming methods for solar tracking(our own and others) against typical dual axis systems. Finally conclusions and future work proposals are made in Chapter 6.

Chapter 2

Background and Related work

2.1 Motivation

2.1.1 Towards a Smart Grid

The existing electricity Grid, in most regions, is a product of various processes connected to economical, political, geographical and other aspects, rather than of mature scientific planning. Even if it had, originally, been a product of scientific planning, the unexpected changes in the production, transmission, conservation and consumption technologies would necessitate a modern shift. The evolution to a Smarter Grid is nowadays imminent, and concerns one of the greatest engineering challenges of our time. In general terms, the move to a Smarter Grid is considered as the move from a centralized, producer-controlled network to a less centralized and more consumer interactive one [1]. The main reasons that call for a radical reengineering of the Grid infrastructure and functionality include the growing demand caused by the electrification of transport and heating and the growing penetration of inherently intermittent and potentially distributed WDERs into the Grid. In the context of power system operation, one of the greatest challenges is running a reliable supply-on-demand system. Historically this challenge led to an electricity Grid based on highly controllable supply in order to match a largely uncontrolled demand [2]. That said, the growing penetration of WDERs into the Grid will impact the systems reliability [3, 4]. Therefore, one of the major challenges of the Smart Grid vision is the reliable integration of WDERs into the Grid while meeting the constantly growing demand.

Predictive technology could reinforce the reliable integration of WDERs into the electricity Grid as "forecasts of future requirements are essential to be able to prepare the controllable and exible systems, such as those based on fossil fuels, to behave in the appropriate manner" [2, 4, 5].

More recently, Artificial Intelligence (AI) and Multiagent Systems (MAS) research has been increasingly involved with building intelligent systems for the Smart Grid [6]. In the process, the efficient incorporation of WDERs into the Smart Grid has emerged as a major challenge [2, 7]. The term Virtual Power Plants (VPPs) corresponds to the notion of a large number of heterogeneous Distributed Energy Resources (DERs), usually WDERs, joining forces and offering electricity to the Grid - while providing the guarantees of a single conventional power plant. VPPs create the necessary synergies among DERs, so that the effective and efficient delivery of energy is assured, while still being able to utilize (the inherently intermittent and thus untrustworthy) WDERs [6, 8, 9]. In such an evolving environment, WDER power production enhancement and dependable power output predictive technology can be considered as a priori requirements.

2.1.2 RENES: A WDERs power output prediction tool

The algorithmic framework for a photovoltaic (PV) system and wind generator power output estimator is presented in [10] along with an interactive web-based tool, *RENES*¹. A generic method to produce PV power output estimates is described based on a solar irradiance approximation model that takes cloud coverage into account, incorporates free-to-use and readily available meteorological data, and satisfies specific performance guarantees for a wide region of interest. This solar irradiance model is built with components chosen after being carefully evaluated against each other in a broad geographic region-the Mediterranean belt (Med-Belt). The components in question are non-linear approximation methods for turning cloud-coverage into radiation forecasts, such as an MLP neural network with one hidden layer. Moreover, the specific method makes use of online data that can be downloaded for free from weather forecasting websites, and do not rely on complex and expensive weather models and data. Hence the presented method can be used as a generic but low-cost power output estimation tool which is applicable within a wide geographical region. Also it is demonstrated there how standard machine learning methods, like least-squares fitting and neural networks, can be effectively applied to predict the power output of solar plants in a wide region. It is important to mention that the use of "intermediate steps", such as using a solar irradiance model, allows this method to be applicable outside narrow regions as would be the case if the neural network was trained just over specific plants' production output data.

¹<http://www.intelligence.tuc.gr/renes/>

2.2 Photovoltaic and solar tracking technology

2.2.1 Solar radiation model

In general, the total incident irradiance G_T falling on an arbitrarily oriented surface, consists of the *beam* G_B , *sky-diffuse* G_D and *ground-reflected* G_R components [11]:

$$G_T = G_B + G_D + G_R \quad (2.1)$$

Usually, the cosine effect is used to model the variations of the G_B component:

$$G_B = G_B^{max} \cos \theta_s \quad (2.2)$$

where

θ_s , is the angle between the normal to the surface and the direction to the sun,

G_B^{max} , is the incident beam irradiance when the surface is oriented normally to the incoming radiation. G_B^{max} is the maximum beam irradiance that the PV module can orient to.

Next, the G_D component formula varies and is based on the assumption that every point of the celestial sphere emits light with equal radiance [12]:

$$G_D = G_D^{max} \frac{1 + \cos \beta}{2} \quad (2.3)$$

where

β , is the inclination angle of the surface,

G_D^{max} , is the incident diffuse irradiance on a horizontally oriented surface. G_D^{max} is the maximum diffuse irradiance that the PV module can orient to.

Finally, the G_R component is formed assuming that the ground is horizontal, of infinite extent, and reflects uniformly to all directions [11]:

$$G_R = G_R^{max} (1 - \cos \beta) \quad (2.4)$$

where

G_R^{max} , stands for the reflected incident irradiance for a 90° surface slope angle. G_R^{max} is the maximum reflected irradiance that the PV module can orient to.

2.2.2 Photovoltaic operational principles

PVS have become increasingly popular during the past years because of their renewable nature and the low-to-zero carbon dioxide emissions that they produce during operation. PVS depend on solar panels, formed basically from solar cells put together. The working principle of all today solar cells is essentially the same. It is based on the photovoltaic effect. In general, the photovoltaic effect means the generation of a potential difference at the junction of two different materials in response to visible or other radiation. The basic processes behind the photovoltaic effect are [13]:

- generation of the charge carriers due to the absorption of photons in the materials that form a junction,
- subsequent separation of the photo-generated charge carriers in the junction
- collection of the photo-generated charge carriers at the terminals of the junction.

2.2.3 PVS tracking solutions

PVS is a functional arrangement of many different components designed to supply electric power using the solar power the way described above. The components of these systems usually are:

- The solar panels, solar cells assembled together in a rectangular metal frame with a plastic or glass surface, forming a weatherproof construction. A linkwired collection of solar panels is named solar array.
- The mounting system, the construction that supports the whole system based on the site's specificity(ground,building roof). A set of intersected metal bars acts as a platform for the solar panels to be mounted onto. In cases of moving(tracking) systems a strong, underground foundation is used to provide the desired stability against the inertial forces produced by the rotating parts.
- The solar tracker, which is the mechanism that rotates the solar panels during the daylight period. Different types of trackers exist depending on the tracking algorithm(astronomical, sensor-based feedback) and the number of axes involved in movement(single axis, dual axis).
- The inverters, electrical systems designed to convert the DC power produced from the solar panels to AC power for public power grid usage.

For the general purposes of this thesis is essential to focus more on the solar tracking implementations. Two main categories can be found regarding the tracking principle:

- *Passive trackers*, as seen in 2.1, are based on thermal expansion of a matter (usually Freon) or on shape memory alloys. Usually this kind of tracker is composed of couple of actuators working against each other which are, by equal illumination, balanced. By differential illumination of actuators, unbalanced forces are used for orientation of the apparatus in such direction where equal illumination of actuators and balance of forces is restored. Passive solar trackers, compared to active trackers, are less complex but work in low efficiency and at low temperatures they stop working. Also they present a higher positioning delay and for these reasons they have not yet been widely accepted by consumers.
- *Active trackers* can be categorized as microprocessor and electro-optical sensor based, PC controlled date and time based, auxiliary bifacial solar cell based and a combination of these three systems.

In this work we use active trackers model because of our intention to supply the PVS with positioning strategies calculated in PCs. Active trackers can be divided in two main categories regarding the number of moving axis.

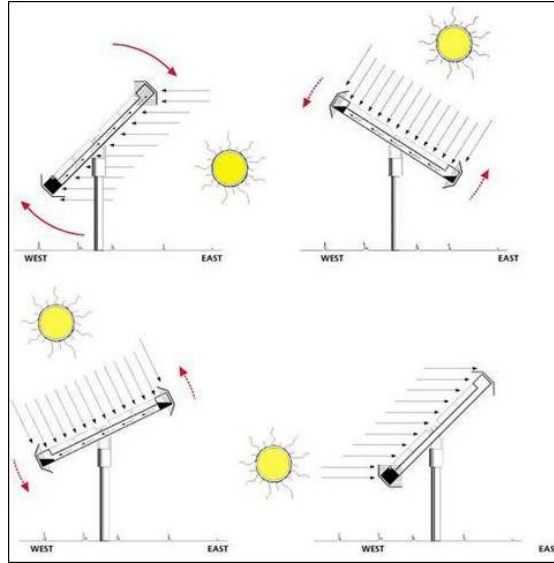


FIGURE 2.1: Passive Tracker

Single axis tracking In this case the system is moving across the north-south(2.2) or the east-west direction(2.3) following the sun's vertical or horizontal movement respectively. When moving across the E-W direction the system remains in fixed position in the

north-south direction. Optimal tilt(N-S orientation) angle, S_{opt} , is a topic under research over the last years with site specific properties, like latitude φ , being taken under consideration. Although research community lacks a definite answer, few suggestions have been made like Lunde [14] $S_{opt} = \varphi \pm 15^\circ$, Duffie and Beckman [15] $S_{opt} = (\varphi + 15^\circ) \pm 15^\circ$, Chang [16] $S_{opt} = 0.9 * \varphi$ for latitudes below 65° and $S_{opt} = 56 + 0.4 * (\varphi - 65)$ otherwise, and Lewis [17] $S_{opt} = \varphi \pm 8^\circ$.

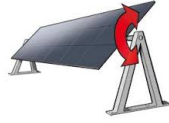


FIGURE 2.2: North-South

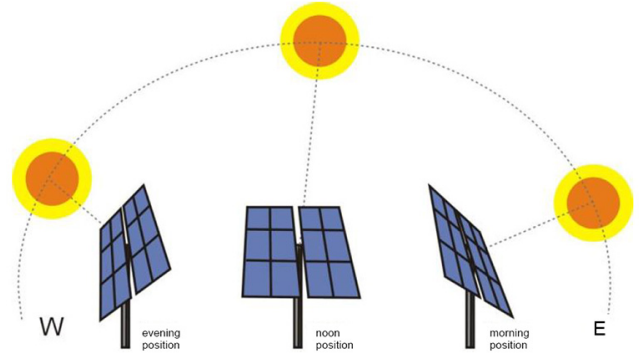


FIGURE 2.3: East-West

Dual axis tracking This is the most advanced configuration in terms of tracking the sun because it allows the vertical movement of the panels in addition to the horizontal one. The Azimuth-altitude dual axis tracker (AADAT) is named after the azimuthal(east-west) and the altitude(north-south) movement of the tracker. Although the additional complexity of the azimuth orientation driving mechanism must be taken into account in terms of system's positioning accuracy, it is reported that even by a 10^0 misalignment, the power output of the panel is still 98.5% of that of perfect position [18].

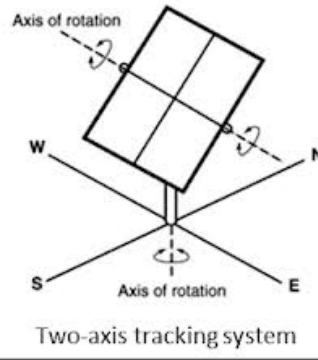


FIGURE 2.4: Dual axis tracker

2.3 Markov Decision Process-MDP

Real-world decision making situations often have one or more of the following characteristics:

- there are multiple and conflicting objectives to be considered,
- decisions are to be made in an environment of uncertainty and risk,
- thought needs to be given to a decision in terms of its impact on the future decision making environment.

Problem solving situations, or portions of them, frequently exist which can be adequately described quantitatively, i.e., the dynamics of the system subject to control, and the criterion, may be expressed mathematically. Sequential decision problems can be formally expressed as MDPs.

Markov Decision Process(MDP) is a 4-tuple (S, A, R, T) [\[19\]](#) which contains:

- a set of states $s \in S$
- a set of actions $\alpha \in A$
- a reward function $R(s, \alpha)$ is the reward received, following action α in state s .
- a transition model $T : P_\alpha(s, s') = Pr(s_{t+1} = s' \mid \alpha_t = \alpha, s_t = s)$ is the probability that action α in state s at time t will lead to state s' at time $t + 1$

Markov decision processes may be classified according to some characteristics: the time horizon in which the decisions are made and the timing of the decisions[\[20\]](#).

With regard to time horizon, finite- and infinite-horizon MDPs are formed. Finite-horizon and infinite-horizon MDPs have different analytical properties and solution algorithms. Because the optimal solution of a finite-horizon MDP with stationary rewards and transition probabilities converges to that of an equivalent infinite- horizon MDP as the planning horizon increases and infinite-horizon MDPs are easier to solve and to calibrate than finite-horizon MDPs, infinite-horizon models are typically preferred when the transition probabilities and reward functions are stationary. However, in many situations, the stationary assumption is not reasonable.

Markov decision processes can be also classified with respect to the timing of the decisions. In a discrete-time MDP, decisions can be made only at discrete-time intervals, whereas in a continuous-time MDP, the decisions can occur anytime. Continuous-time MDPs generalize discrete-time MDPs by allowing the decision maker to choose actions

whenever the system state changes and/or by allowing the time spent in a particular state to follow an arbitrary probability distribution.

In MDPs, we assume that the state the system occupies at each decision epoch is completely observable. However, in some real-world problems, the actual system state is not entirely known by the decision maker, rendering the states only partially observable. Such MDPs are known as POMDPs, which have different mathematical properties than completely observable MDPs and are beyond the scope of this thesis.

The core problem of solving an MDP is to find the policy $\pi : S \times A \rightarrow [0, 1]$, which is a function that specifies the action a to be taken when in state s in order to maximize some long-run cumulative function of reward[21]

$$V^\pi(s) = E\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots | s_t = s, \pi\} \quad (2.5)$$

$$\begin{aligned} &= E\{r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s, \pi\} \\ &= \sum_{a \in A} \pi(s, a) \left[R(s, a) + \gamma \sum_{s'} P_a(s, s') V^\pi(s') \right], \end{aligned} \quad (2.6)$$

where $\pi(s, a)$ is the probability with which the policy π chooses action $a \in A$ in state s , and $\gamma \in [0, 1]$ is a discount factor. This quantity, $V^\pi(s)$, is called the value of state s under policy π , and V^π is called the state-value function for π . The optimal state-value function gives the value of each state under an optimal policy:

$$V^*(s) = \max_{\pi} V^\pi(s) \quad (2.7)$$

$$\begin{aligned} &= \max_{a \in A} E\{r_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a\} \\ &= \max_{a \in A} \left[R(s, a) + \gamma \sum_{s'} P_a(s, s') V^*(s') \right]. \end{aligned} \quad (2.8)$$

Any policy that achieves the maximum in (2.7) is by definition an optimal policy. Thus, given V^* an optimal policy is easily formed by choosing in each state s any action that achieves the maximum in (2.8). Particularly useful to compute value functions and thereby to optimize or improve policies are Bellman equations, such as (2.5), (2.6) which recursively relate value functions to themselves. If the values V^* or V^π , are treated as unknowns, then a set of Bellman equations, for all $s \in S$, forms a system of equations whose unique solution is in fact V^π or V^* as given by (2.5) or (2.7) respectively. Given that, V^* forms a non-linear set of equations, due to the \max operator, excluding this way linear programming approaches as infeasible. In order to estimate an optimal solution for Bellman equations dynamic programming techniques are used.

MDPs are widely used in several fields such as inventory management, communication

models and systems networking, behavior ecology and even in gambling! As an application example we can see the inventory management and the goal of optimal reorder points and reorder levels. The time that decision is made is every week and the state space is product inventory level at the time of the decision. The actions available are the amount of stock to order and the transition probability is how much stock ordered and the random demand for that week. A decision rule specifies the quantity to be ordered as a function of the stock on hand at the time of decision. A policy consists of a sequence of such restocking functions.

For our problem presented in this work the state space will be relevant to orientation positions of the PVS and the reward model will describe the estimated energy produced in a certain position minus the energy consumed from the system in order to reach this position.

Dynamic programming is an optimization approach that transforms a complex problem into a sequence of simpler problems, the subproblems, and relates their solutions. Dynamic programming provides a general framework for analyzing many problem types. Within this framework a variety of optimization techniques can be employed to solve particular aspects of a more general formulation. Sometimes insight is required in order to recognize that a particular problem can be handled effectively as a dynamic program or to restructure the problem's formulation so that it can be solved effectively. Problems of this family present some common characteristics[22], which can be described as the stages and the states of the problem along with a recursive optimization procedure.

Stages can be seen as the structural units of optimization problems, which are solved sequentially one stage at a time. Although each one-stage problem is solved as an ordinary optimization problem, its solution helps to define the characteristics of the next one-stage problem in the sequence. Sometimes the stages represent different timesteps in the problem's time horizon and other times they do not have time implications.

The states of the process are associated with each stage of the optimization problem. The states reflect the information required to determine the consequences of a current decision upon future actions. The specification of the system's states is perhaps a crucial design parameter of the dynamic programming model. No standard rules exist for this construction but some generic properties are adopted. The first is that enough information should be propagated through states to make future decisions and the second is that the number of state variables should be small enough, since the computational cost would be prohibitively large otherwise.

The last general characteristic of the dynamic programming approach is the development of a recursive optimization procedure, which builds a solution for the multi-stage

problem by first solving a one-stage problem and sequentially using this result to solve the next one-stage problems until the overall optimum has been found. This procedure is usually based on a backward or forward induction process, where the first stage to be analyzed is the final stage of the problem and problems are solved moving back one stage at a time or the first stage to be solved is the initial stage of the problem and problems are solved moving forward one stage at a time until all stages are included. The basis of the recursive optimization procedure is the *principle of optimality*, which states that *an optimal policy has the property that, whatever the current state and decision, the remaining decisions must constitute an optimal policy with respect to the state resulting from the current decision*.

In our case acting optimally at each timestep, which is the decision epoch of our problem, will result in acting optimally through the whole day, until all timesteps have finished.

Based on the above there are 2 fundamental dynamic programming algorithms for solving MDPs: *value iteration* and *policy iteration* methods.

Value iteration[23] starts with an arbitrary value for each state and, at each iteration, solves equation (2.8) using the value from the previous iteration until the difference between successive values becomes sufficiently small (stopping criterion ϵ). The value corresponding to the decision maximizing equation (2.8) is guaranteed to be within a desired distance from the optimal solution. Comparing this algorithm with dynamic programming guidelines certain relations are recognised. States are used and the iterations correspond to time stages. Finally, by solving recursively the Bellman equation for each state, corresponding to the subproblem mentioned above, an overall optimal approximation is derived.

Algorithm 1 Value Iteration

```

procedure VALUEITERATION
  for all  $s \in S$  do
     $V_0(s) = 0$  ▷ arbitrary initial value
  end for
  while  $\forall s \in S, |V_{k-1}(s) - V_k(s)| > \epsilon$  do ▷  $\epsilon$  :stopping criterion
    for all  $s \in S$  do
      for all  $a \in A$  do
         $V_k(s) \leftarrow R(s, a) + \gamma \max_a \sum_{s'} P_a(s, s') V_{k-1}(s')$ 
      end for
       $\pi_k^*(s) \leftarrow \arg \max_a V_k(s)$ 
    end for
  end while
  return  $\pi_k^*(s), V_k(s)$ 
end procedure

```

Policy iteration[24] starts with an arbitrary decision rule and finds its value; if an improvement in the current decision rule is possible, using the current value function estimate, then the algorithm will find it; otherwise, the algorithm will stop, yielding the optimal decision rule.

Algorithm 2 Policy Iteration

procedure POLICYITERATION

$k \leftarrow 0$

$\pi_0 \leftarrow$ arbitrary initial policy

repeat

$k \leftarrow k + 1$

 solve $V^{\pi_{k-1}}(s) = R(s, \pi_{k-1}(s)) + \gamma \sum_{s'} P^{\pi_{k-1}(s)}(s, s') V^{\pi_{k-1}}(s')$ ▷ Policy Evaluation

for all $s \in S$ **do**

$\pi_k(s) \leftarrow \arg \max_a R(s, a) + \gamma \sum_{s'} P^a(s, s') V^{\pi_{k-1}}(s')$ ▷ Policy Improvement

end for

until $\pi_k = \pi_{k-1}$

return π_k

end procedure

As for the dynamic features of the algorithm policy iteration finds the value of a policy in a recursive manner by applying the backward induction algorithm while ensuring that the value functions for any 2 subsequent stages are identical. Finally the policy produced is optimized until policies of the same state for successive steps have converged.

2.4 Related Work

As stated in the beginning of this thesis dynamic programming approaches in sun tracking have never been tried before for certain reasons. Thus it was chosen to present, as related literature, attempts of purely heuristic nature but with the same goal; PVS efficiency improvement.

In [25] it is presented a control application of a sun tracker that is able to follow the sun with high accuracy without the necessity of either a precise procedure of installation or recalibration. A hybrid tracking system that consists of a combination of open loop tracking strategies based on solar movement models and closed loop strategies using a dynamic sensor feedback controller. Energy saving factors are taken into account, which implies that, the sun is not constantly tracked with the same accuracy, to prevent energy overconsumption by the motors. The system utilizes a hybrid method of tracking the sun, a combination of normal tracking mode and a search mode. Specifically in normal

tracking mode, system adopts open loop tracking strategies based on solar movement models (feedforward control) and closed loop control strategies using a feedback controller. The feedback controller is designed to correct the tracking errors made by the feedforward controller in the open loop mode. For energy consumption reasons, the tracker does not move as long as the tracking error (assuming that the sun is where it is supposed to be) is less than a certain threshold. When this error is greater than this threshold, the controller orders the driver to move to a point at which the sun will arrive in a certain amount of time. While the sun tracking error is small enough and the solar radiation great enough for the system to produce electric energy, the previous strategy is followed. Otherwise the search mode is initialized, the movement of the PVS follows a square spiral in the azimuth-elevation plane in order to detect the position of the sun. As the movement takes place, a check is made as to whether the system is generating electric power. As soon as electric power is produced, this mode is abandoned, and the controller enters the normal tracking mode.

A similar technique, making use of sensor feedback and open loop methods, was developed in [26]. When the system starts, the tracker is set to the home position and then takes GPS information to calculate the sun set and rise times. The present solar time is compared with the sunrise and sunset times to determine whether tracking should start or stop. At night time, it waits next sample time. Sample time period is defined according to technical constraints, concerning GPS hot start time that is 1 second and the energy consumed by motors during one tracking step. When present solar time belongs between sunrise and sunset times, control system acquires pyranometer readings to check if there is enough solar radiation to generate power. Otherwise, sun tracker stays at home position until solar radiation rise to higher limits. After solar radiation reaches the desired value, algorithm reads anemometer value to define whether the sun tracker can move safely. If not, sun tracker stays at home position at least during one sample time. Otherwise, it starts tracking the sun.

An approach employing a sliding mode control method of self-optimization is presented in [27]. The PVS rotates its dual axes by modifying the azimuth angle and the elevation angle accordingly until the panel's power output value reaches maximum. By relating the power output to the azimuth and elevation angles, the system steadily rotates its motors until the power produced by the solar panel converges towards the expected optimum, stated by the manufacturer. Afterwards oscillates within a neighborhood of the optimum point until some error threshold is approached. As a result this methodology manages for the system to operate in maximum power output and avoid the use of complex astronomical equations to find the sun angles.

The main objective of [28] is to determine the orientation angles values which maximize the received direct radiation on an azimuthally tracked PV panel and minimize the total energy consumption of the motors involved, for a given steps number $n = 0, \dots, 4$.

For energy saving reasons, the angular displacement of the tracked PV panel is made discontinuously (stepwise). In order to achieve this objective, firstly, an analysis of the existing angular systems in which the sun's path can be described, is made, presenting the corresponding correlations. Using the previous correlations the unit vectors of the sun-ray and of the normal to the panel are calculated. These unit vectors are needed for the calculus of the incidence angle between the PV panel normal and the sun-ray; on the basis of the incidence angle, the received direct solar radiation is established.

Another solar tracking system is proposed based on the intensity of solar radiation which is the most basic input parameter[29]. Establishing a suitable model of solar radiation is essential for the design of the solar tracking system. The sun elevation angle marked as a is, therefore, calculated. The solar radiation received by the solar cell array mainly depends on the solar incidence angle θ , while solar incidence angle marked as θ is a function of sun equatorial latitude angle δ , local dimension ϕ , solar cell array tilt angle β , solar cell array azimuth angle γ and solar hour angle ω . Their specific relationship is also calculated and according to this, specific solar incidence angle θ of the solar cell array in any location, season, time and geometric position is available. Also a hardware implementation based on the above is presented at the end.

Furthermore in another approach, the trajectories of the sun tracking system are determined in an optimization procedure[30]. The optimization goal is maximization of an electric energy production in the photovoltaic system considering the tracking system consumption. Determination of the tilt angle and azimuth angle trajectories is described as a nonlinear and bounded optimization problem, where the objective function is not available in the explicit form. A *stochastic search* algorithm called Differential Evolution is used as an optimization tool. In the optimization procedure, the objective function is evaluated by using the models of available solar radiation, tracking system consumption, and the efficiency of solar cells with the appropriate dc/dc converters. The problem bounds are given in the form of lower and upper bounds for both angles and time and angle quantization.

The idea of designing and optimizing a tracking mechanism, which changes the position of the PV panel in order to maximize the solar radiation degree of use is met in[31]. The tracking system is approached in mechatronic concept, by integrating the electronic control system in the mechanical structure of the solar tracker. In order to accomplish this goal, it is used an optimization strategy which intends to optimize the motion/control law, in fact to obtain as much as possible energy gained through tracking the sun path with minimum energy consumption for driving the system. The key idea for optimizing the motion law of the panel is to maximize the energy gained through orientation with minimum energy consumption for driving the system. The photovoltaic panel can be rotated without brakes during the day-light, or can be discontinuously driven (step-by-step motion), usually by rotating the panel with equal steps at every hour. It is stated

that the maximum incoming radiation is obtained for the continuous orientation of the panel/system, facing east in the morning and west in the afternoon, in the entire angular field, $\beta^* \in [-90^\circ, +90^\circ]$, but in this case the operating time of the motor, and consequently the energy consumption, is high (β^* is the daily angle of the panel). The tracking cases were formulated using optimal algorithms based on the angular field of the daily motion. These algorithms were developed considering the correlation between the maximum amplitude of the motion and the operating time of the motor, aiming to the minimization of the energy consumption for tracking the sun. In fact, the optimization is made by reducing the angular field of the revolute axis without significantly affecting the incoming energy.

Another interesting approach for optimal sun tracking was developed as part of the diploma thesis of Giannis Kantaris [32] of the School of Electronic and Computer Engineering in TUC. The main idea of this work is that the system tracks the solar irradiance rather than the sun's movement. In more detail, the PVS absorbs a certain quantity of solar irradiance. Based on this quantity an output current is produced and its value is proportional to the irradiance. The system uses the current value in order to calculate the voltage value through a current sensor. Subsequently a micro-controller reads the voltage value and based on this it decides the system's movement direction by supplying the stepper motors, which rotate the system, with appropriate electric pulses. This procedure repeats until the system absorbs maximum solar irradiance judging from the output voltage values.

Special mention should be made in another work [33] because of its AI-based approach, which was developed in parallel with this thesis. That work proposes a problem-specific policy iteration method (along with two specialized variants), which is able to calculate near-optimal trajectories for efficient day-ahead solar tracking, based on weather forecasts coming from online providers for free. To account for the energy needs of the tracking system, the technique employs a novel and generic consumption model that was put forward and is used by this thesis also. The simulation results show that the proposed methods can increase the power output of a PVS considerably, when compared to standard tracking techniques.

In more detail, the problem is modeled as a fully observable, finite horizon MDP with discrete space and time sets. In order to approximate the optimal solar tracking policy corresponding to the optimal value function V^* as described from Bellman in eq. 2.8, page 10, a few methods are proposed based on policy iteration for dual and single axis systems. The first procedure, named as *Solar Tracking Policy Iteration*, considers an arbitrary input policy for the MDP described above, e.g., a myopic one and then attempts to improve that policy but assuming a fixed azimuthal positions sequence. Given this fixed azimuthal sequence, it computes the respective optimal slope-positioning policy.

The output policy is then used to fix a slope positions sequence and based on this another PI algorithm estimates an optimal azimuth positioning policy. This turn-by-turn intermittent optimization process repeats until convergence, or until some computational or time limit is reached. In this way, by combining the derived sequence of positions corresponding to the policies computed for each axis, a policy specifying a sequence of tracker positioning actions is derived.

The same PI algorithm is employed also for single axis tracking, with the action selection process for the static axis (the slope one, in this case) considering only a set of fixed possible orientations for the whole motion.

Next, a *Myopic method* is introduced as the policy that greedily maximizes power generation only without taking into account the system's consumption. As stated power output of a PVS is monotonically correlated with the incident irradiance, hence eq. 2.1 in page 5 should be maximized. As seen in eqs. 2.3 and 2.4 G_D and G_R components vary from their maximum based only on the slope angle of the PV module while, as eq. 2.2 suggests, the G_B component varies from its maximum based on the incident angle, which for any given sun position depends on the slope and azimuth angle of the PV module. This means that the beam component reaches maximum when the module is ideally aligned with the sun. Thus by fixing the azimuth angle to follow the sun azimuth, is ensured that the maximum G_T is tracked in all cases. Then the only thing left to be optimized is the PVS slope angle at every time step, so that the G_T components are properly balanced and get the maximum G_T possible.

For single axis tracking, the problem is further simplified into following the sun over the azimuth (and just defining the best next-day fixed slope orientation).

At last, serving as a baseline, a next-day-optimal fixed PVS orientation method is proposed by simply searching the whole space of possible orientations, given the weather prediction for the next day.

Chapter 3

Methodology

In this chapter it is presented the methodology that was followed in order to express the problem in a formal way, to describe the datasets and the corresponding days of the year and finally outline the main aspects of the algorithm used.

3.1 Real world problem representation

It is very well known that the transfer of a real world problem in a finite simulated environment is a difficult task especially when computational tractability is required. This goal was achieved by modeling the PVS physical aspects, discretizing the continuous space-time of the problem and updating our aforementioned mathematical tools accordingly.

3.1.1 Physical aspects of the system

A fundamental piece of this work is to model a typical, large scale($\sim 10kW$), commercially available PVS with dual axis tracking capability. The model's accuracy is vital because physical measures will affect energy production along with energy consumption of the examined system. Furthermore good precision will justify simulation results to be used in real applications. After reviewing some commercial solutions [34? , 35] the following numbers were produced:

- length: $12m$
- width: $6m$
- depth: $0.2m$

- mass: $2500kg$.

3.1.2 Discretization of continuous space-time

Space

A key characteristic in PVS is the motion range achieved by the motors across both the horizontal and vertical axis. It is fairly common the movement allowed in tracking systems to be constrained within a certain range in both the azimuthal and elevation axis. The same approach was followed, by reviewing typical, available solutions[34, 36–40], and the resulting numbers are:

- azimuth range, $r_{az} = 270^\circ$
- elevation range $r_{el} = 63^\circ$.

Also stepping motors were chosen to represent the rotating mechanisms. Stepping motors are widely used in such applications, both in research[41–43] and in industry[44]. Some reasons are that they are easily connected with advanced driving controllers, produce high amount of torque required by inertial forces and they consume less energy than a constantly rotating motor. Typical step resolution, θ_{step} , of such motors are 0.9° and 1.8° , although the first is avoided due to the complicated electrical circuits that requires. Thus $\theta_{step} = 1.8^\circ$ was picked for our model. Based on this choice, motion ranges can be re-expressed as sets of discrete points:

- azimuth set $|K| = \frac{r_{az}}{\theta_{step}} + 1 = \frac{270^\circ}{1.8^\circ} + 1 = 151$
 $K = [1, 2, \dots, 151]$
- elevation set $|\Lambda| = \frac{r_{el}}{\theta_{step}} + 1 = \frac{63^\circ}{1.8^\circ} + 1 = 36$
 $\Lambda = [1, 2, \dots, 36]$.

Time

Assuming step by step movement control, an interval $\Delta_{intrv} = 5min$ was established between consecutive control activations. Because of the varying duration of daylight, denoted by D_{day} , throughout the year, the number of timesteps, τ , included from sunrise to sunset should be expressed in lower-upper limits form. For our site of interest, Chania, the length of daylight for summer solstice(21 June) and winter solstice(21 December), biggest and smallest day respectively, was retrieved¹. At first a set I of timesteps τ is

¹http://www.sunearthtools.com/dp/tools/pos_sun.php?lang=en

introduced as follows

$$|I| = \frac{D_{day}}{\Delta_{intrv}} + 1 \quad (3.1)$$

1. **21 June**= 14:33 hours of daylight

$$\begin{aligned} |I| &\leq 14 * \frac{60min}{\Delta_{intrv}} + \frac{33min}{\Delta_{intrv}} + 1 \iff \\ |I| &\leq 175 \end{aligned} \quad (3.2)$$

2. **21 December**= 9:45 hours of daylight

$$\begin{aligned} |I| &\geq 9 * \frac{60min}{\Delta_{intrv}} + \frac{45min}{\Delta_{intrv}} + 1 \iff \\ |I| &\geq 118 \end{aligned} \quad (3.3)$$

Combining 3.2,3.3 results in:

$$118 \leq |I| \leq 175 \quad (3.4)$$

All of the above results gathered in a tabular form

length	width	depth	mass
12m	6m	0.2m	2500kg

TABLE 3.1: Physical dimensions

	azimuth range	elevation range	time
continuous	270°	63°	9 : 45 ~ 14 : 33(h)
discrete($\theta_{step} = 1.8^\circ, \Delta_{intrv} = 5min$)	151	36	$118 \leq I \leq 175$

TABLE 3.2: Space-time characteristics

3.1.3 MDP representation

Based on the results from previous subsections 3.1.1 ,3.1.2 an update of our MDP mathematical framework and its proposed solving method is made. As described in 2.3 MDP is an (S, A, R, T) tuple and its components should be updated respectively.

State space $\equiv \mathbf{S}$

\mathbf{S} is a finite set of states, consisting of all possible azimuth orientation positions in the set K , all possible elevation orientation positions in the set Λ and all possible controller interaction timesteps, τ , in the set I combined together. So every state $s \in \mathbf{S}$ is a

$(\kappa_s, \lambda_s, \tau_s)$ tuple, with $\kappa_s \in [1, |K|]$, $\lambda_s \in [1, |\Lambda|]$, $\tau_s \in [1, |I|]$. As a result

$$|S| = |K \times \Lambda \times I| \quad (3.5)$$

Action space $\equiv \mathbf{A}$

A is a finite set of actions, with each action $\alpha \in A$ positioning the system in a specific azimuth-elevation orientation. So α is a (κ_s, λ_s) tuple, with $\kappa_s \in [1, |K|]$, $\lambda_s \in [1, |\Lambda|]$ and A resulting in

$$|A| = |K \times \Lambda| \quad (3.6)$$

Reward function $\equiv R(s, \alpha, s')$

Reward model $R(s, \alpha, s')$ is a function that calculates reward values for transitions from state s to state s' following action α . In the current work, rewards correspond to energy produced from the PVS between two consecutive timesteps. Also the energy consumption of the PVS, as described in [33], during the same period was taken into account. Hence

$$R(s, a, s' | \tau_s, \tau_{s'}) = Prod(s, s') - Cons(s, s') \quad (3.7)$$

The $Prod(s, s')$ function calculates the estimated energy produced as the PVS follows the orientation pattern every Δ_{intrv} .

$$Prod(s, s') = \Delta_{intrv} * \frac{Pwr_{est}(s) + Pwr_{est}(s')}{2} \quad (3.8)$$

where $Pwr_{est}(\bullet)$ represents the estimated power output of the PVS at each state s . These values are calculated from RENES backend. The given input is the dataset file as described in section 3.2.3 page 33 and the output are values in a tabular form as presented in table 3.2.

The $Cons(s, s')$ function calculates the power needed by the PVS for the transition from s to s' , taking as inputs numerous variables regarding the weather conditions and the physical measurements of the modeled system.

Transition model $\equiv T = P_\alpha(s, s')$

The transition function $P_\alpha(s, s')$ determines the possibility that taking action α in state s will lead in state s' . Because in this case the PVS rotating motors can be electronically controlled, movements are considered deterministic. Thus

$$P_\alpha(s, s') = Pr(s_{t+1} = s' | \alpha_t = \alpha, s_t = s) = 1 \quad (3.9)$$

updated MDP(S,A,R,T)

- state space $S : |S| = |K \times \Lambda \times I|$
- action space $A : |A| = |K \times \Lambda|$
- reward model $R(s, a, s' | \tau_s, \tau_{s'}) = Prod(s, s') - Cons(s, s')$
- transition model $T : P_\alpha(s, s') = Pr(s_{t+1} = s' | \alpha_t = \alpha, s_t = s) = 1$
- Bellman equation as in 2.8 page 10

$$\begin{aligned}
 V^*(s) &= \max_{a \in A} \left[R(s, a) + \sum_{s'} P_a(s, s') V^*(s') \right] \\
 &= Prod(s, s') - Cons(s, s') + \max_{a \in A} \sum_{s'} V^*(s')
 \end{aligned} \tag{3.10}$$

3.2 Datasets

In this section the reasons for selecting specific days of the year are explained, also some astronomical aspects of these days are calculated and finally the way these selected days were transformed into datasets based on observed weather data acquired online.

3.2.1 Reasoning of selected dates

As seen in 3.2 the two solstices are the extremes, regarding the daylight length, inside the year's cycle. Also these days represent different seasons(winter-summer) and weather conditions that occur during each season. Hence winter solstice of *21stDecember* and summer solstice of *21stJune* constitute the first two study cases for the experimental evaluation to come. During the transition from one solstice to the other, another two important days are met; spring and fall equinox. Like solstices, considered as extremes, the two equinox can be seen as the average, because day time and night time are almost equal. Furthermore they belong to the remaining two seasons, spring equinox *20thMarch* and fall equinox *22ndSeptember*, including their weather characteristics. Thus, the days of interest are:

- *21December*, winter solstice

- 21*June*, summer solstice
- 20*March*, spring equinox
- 22*September*, fall equinox.

In order to provide some insight about the cloud coverage of the selected days, historical data, from the past 5 years for each day, were retrieved from online free sources² and are presented below. Only cloud coverage conditions for the aforementioned days was chosen to be presented because of the major effect cloudiness has on solar radiation[45, 46]. The following table summarizes the various observable sky conditions along with their corresponding quantitative expression as found in [47].

Reported	Contraction	Meaning	Summation	Amount of Layer
	CLR	Clear		0
	FEW	Few		1/8 – 2/8
	SCT	Scattered		3/8 – 4/8
	BKN	Broken		5/8 – 7/8
	OVC	Overcast		8/8

TABLE 3.3: Sky cover

For a more compact and comprehensive presentation the data are grouped in 3-hour intervals from sun rise to sun set.

21 December

Year/Time	6 – 9	9 – 12	12 – 15	15 – 18	18 – 21
2012	BKN	BKN	SCT	BKN	BKN
2011	BKN	BKN	BKN	BKN	SCT
2010	SCT	BKN	BKN	BKN	BKN
2009	SCT	SCT	SCT	FEW	SCT
2008	OVC	BKN	BKN	BKN	OVC

TABLE 3.4: 5-year sky cover history for 21 December

Note: During 18-21 pm sky conditions are of none usefulness, other than the archival, because the sun has already set but they are presented for uniformity reasons.

21 June

²<http://www.wunderground.com/>

Year/Time	6 – 9	9 – 12	12 – 15	15 – 18	18 – 21
2012	CLR	CLR	CLR	CLR	CLR
2011	CLR	CLR	CLR	CLR	CLR
2010	CLR	CLR	CLR	FEW	FEW
2009	FEW	CLR	CLR	FEW	FEW
2008	CLR	CLR	CLR	CLR	CLR

TABLE 3.5: 5-year sky cover history for 21 June

22 September

Year/Time	6 – 9	9 – 12	12 – 15	15 – 18	18 – 21
2012	SCT	FEW	FEW	FEW	CLR
2011	SCT	SCT	BKN	BKN	BKN
2010	BKN	BKN	BKN	BKN	BKN
2009	SCT	SCT	SCT	SCT	FEW
2008	OVC	BKN	BKN	SCT	SCT

TABLE 3.6: 5-year sky cover history for 22 September

20 March

Year/Time	6 – 9	9 – 12	12 – 15	15 – 18	18 – 21
2012	CLR	FEW	FEW	FEW	FEW
2011	SCT	SCT	BKN	OVC	BKN
2010	FEW	FEW	FEW	CLR	CLR
2009	CLR	CLR	FEW	FEW	FEW
2008	BKN	BKN	BKN	OVC	OVC

TABLE 3.7: 5-year sky cover history for 20 March

Based on the data above some quality characteristics can be extracted, regarding the sky conditions appearing at each case.

- During summer solstice mostly clear sky occurs over time and only a few cloud cover appears in a sole year.
- In the winter solstice heavy cloud cover is observed through the years, varying from scattered and mainly broken clouds even to complete overcast, in contrast with summer solstice.
- In the case of fall equinox cloud cover is always present with transitional patterns being developed in some of the examined years.
- Similar characteristics can be found in spring equinox also with the sky altering from clear to mildly cloudy along with transitions from heavy to complete cloud cover.

- As for the two equinoxes the presence of all conditions justifies the given attribute "average case".
- Moreover, the transitions in cloud cover during equinoxes describe the general weather conditions' mutability of the corresponding seasons.
- In contrast, winter and summer show specific weather and cloud cover patterns.

3.2.2 Astronomical aspects of the selected dates

For the selected days, we perform a methodical approach, exploiting astronomical information in order to calculate sun's movement and speed in both PVS axes at the location of interest, Chania. The results will be used later in order to define size parameters in our moving windows. During the solstices sun's celestial path reaches its maximum and minimum in both axes of movement whereas during equinoxes average values are observed.

21 December

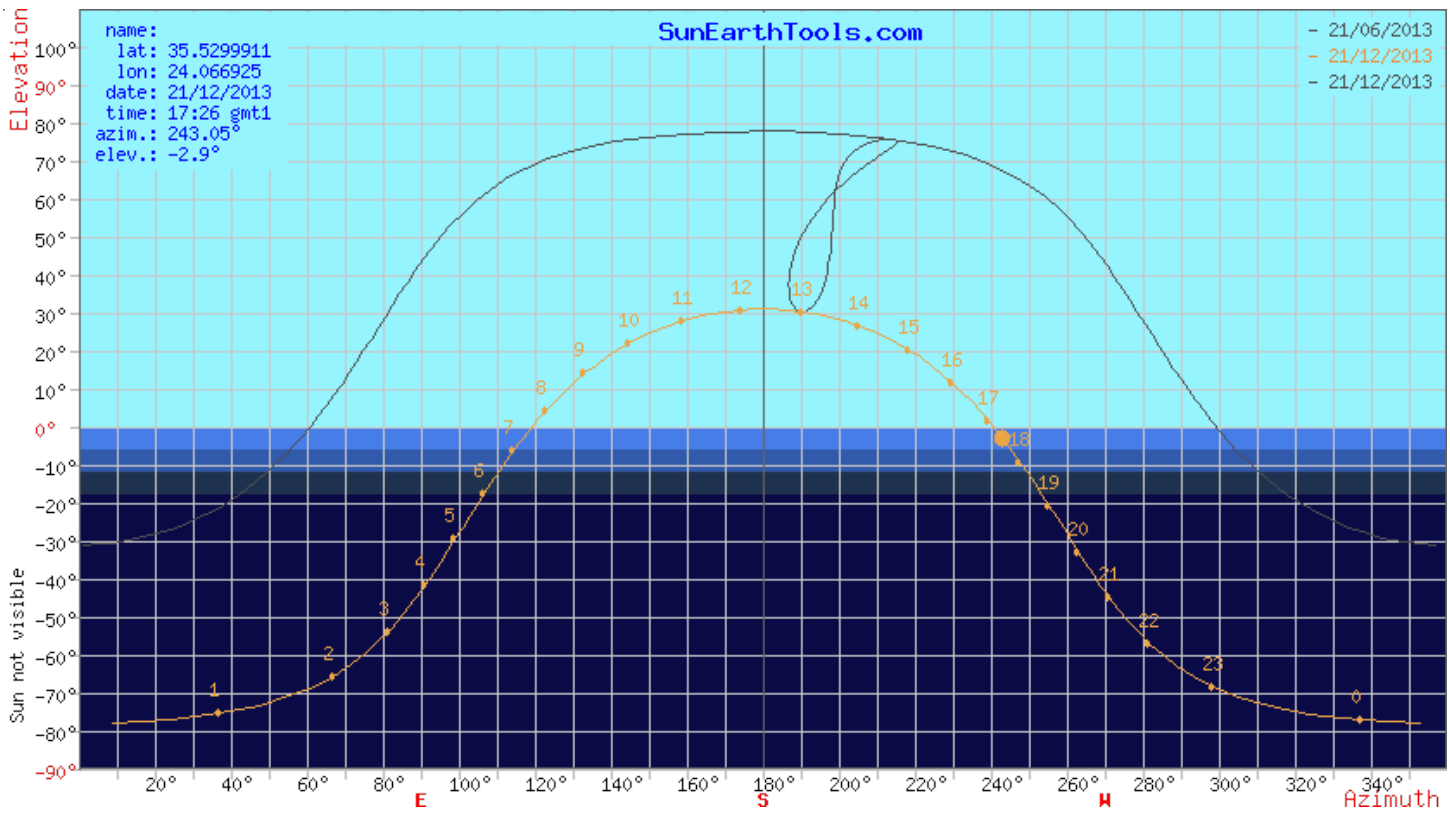


FIGURE 3.1: Cartesian-like sun route for 21 December

- Rise: $\{t_{rise} = 7.30 \Rightarrow \theta(t_{rise}^{az}) = 118.6^\circ, \theta(t_{rise}^{el}) = 0^\circ\}$
- Set: $\{t_{rise} = 17 : 15 \Rightarrow \theta(t_{set}^{az}) = 241.4^\circ, \theta(t_{set}^{el}) = 0^\circ\}$
- Maximum elevation, $\theta_{el}^{max} = 31.1^\circ, \theta(t^{az}) = 180^\circ\}$ at $\sim 12 : 20$
- Daylight duration, $D_{day} = 9.45h$
- Azimuth range, $r_{az} = 122.8^\circ$
- Elevation range, $r_{el} = 2 * \theta_{el}^{max} = 62.2^\circ$

Average speed of azimuth displacement, denoted as \bar{v}_{az} , and elevation, \bar{v}_{el} , are calculated.

$$\bar{v}_{az} = \frac{r_{az}}{D_{day}} = \frac{122.8^\circ}{585min} = 0.21^\circ/min \quad (3.11)$$

and

$$\bar{v}_{el} = \frac{r_{el}}{D_{day}} = \frac{31.1^\circ * 2}{585min} = 0.11^\circ/min \quad (3.12)$$

Sun's speed is not constant during time as seen in 3.1, therefore worst cases(maximum speed) should be also considered for both axes. From 3.1 is observed that azimuthal worst case appears around noon while reaching maximum elevation and elevation worst case is at sunrise and sunset. Also a reasonable and representing interval, $\Delta t = t_2 - t_1 = 20min$ was selected, where t_1, t_2 are time points that meet our worst cases observations.

azimuth

$$t_1^{az} = 12 : 10 \Rightarrow \theta(t_1^{az}) = 176.7^\circ$$

$$t_2^{az} = 12 : 30 \Rightarrow \theta(t_2^{az}) = 182.1^\circ$$

,where $\theta(t^{az})$ is the specific azimuth position at time t_{az} .

So azimuth max speed, v_{az}^{max} , equals:

$$v_{az}^{max} = \frac{\theta(t_2^{az}) - \theta(t_1^{az})}{t_2^{az} - t_1^{az}} = \frac{5.4^\circ}{20min} = 0.27^\circ/min \quad (3.13)$$

elevation

The same applies in elevation also with one addition; because maximum elevation speed occurs both in sunrise and sunset their average angular displacement, $avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el}))$,

is used as nominator.

$$\underbrace{\text{rise} \begin{cases} t_1^{el} = 7 : 35 \Rightarrow \theta(t_1^{el}) = 0^\circ \\ t_2^{el} = 7 : 55 \Rightarrow \theta(t_2^{el}) = 3.6^\circ \end{cases} \quad \text{set} \begin{cases} t_1'^{el} = 16 : 50 \Rightarrow \theta(t_1'^{el}) = 3.6^\circ \\ t_2'^{el} = 17 : 10 \Rightarrow \theta(t_2'^{el}) = 0^\circ \end{cases}}_{(3.14)} \quad (3.14)$$

$$\text{avg}(\Delta\theta(t^{el}), \Delta\theta(t'^{el})) = \frac{(\theta(t_2^{el}) - \theta(t_1^{el})) + (\theta(t_1'^{el}) - \theta(t_2'^{el}))}{2} = 3.6^\circ \quad (3.15)$$

Hence, elevation max speed, v_{el}^{max} , equals:

$$v_{el}^{max} = \frac{\text{avg}(\Delta\theta(t^{el}), \Delta\theta(t'^{el}))}{\Delta t^{el}} = \frac{3.6^\circ}{20min} = 0.18^\circ/min \quad (3.16)$$

21 June

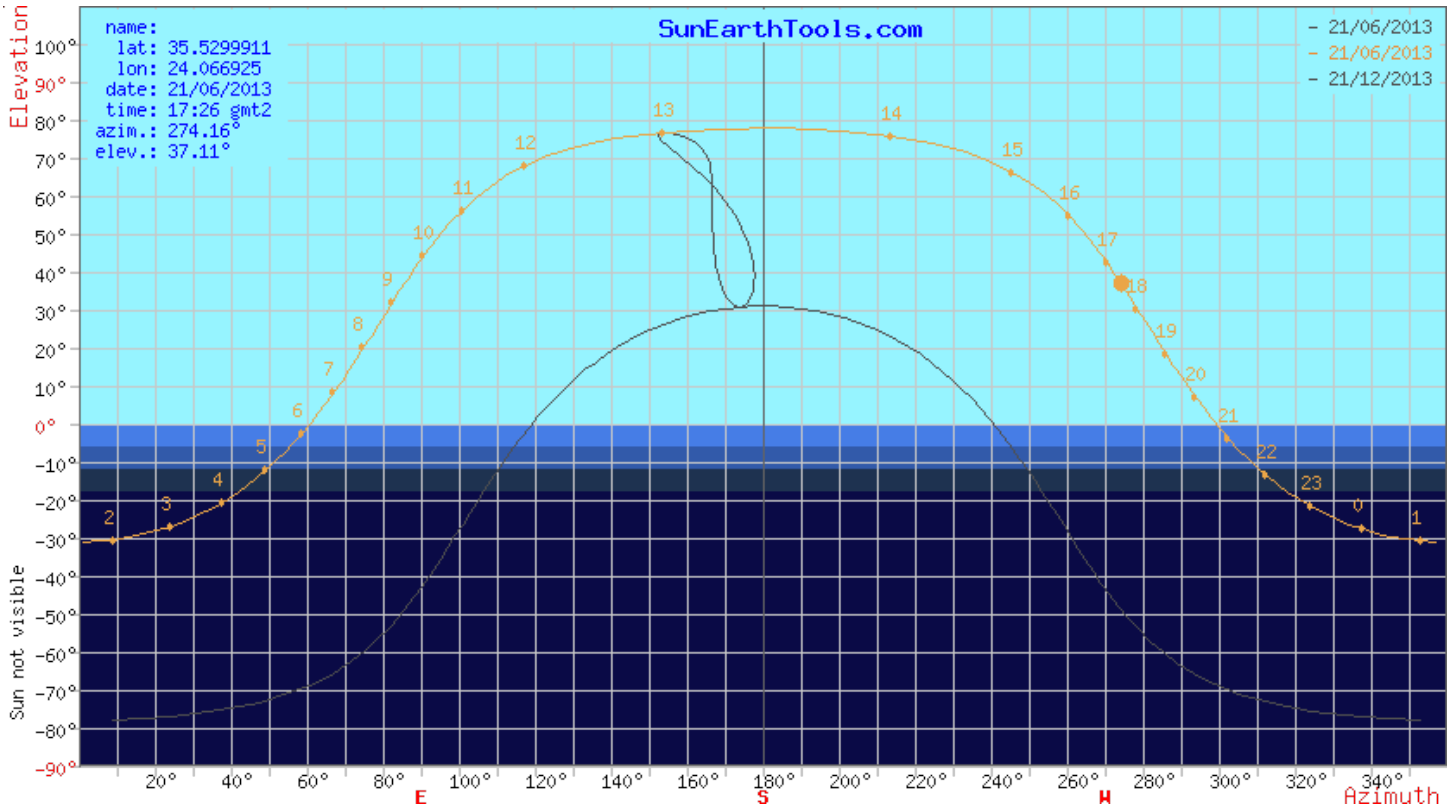


FIGURE 3.2: Cartesian-like sun route for 21 June

- Rise: $\{t_{rise} = 6 : 10 \Rightarrow \theta(t_{rise}^{az}) = 60^\circ, \theta(t_{rise}^{el}) = 0^\circ\}$

- Set: $\{t_{rise} = 20 : 43 \Rightarrow \theta(t_{set}^{az}) = 300^\circ, \theta(t_{set}^{el}) = 0^\circ\}$
- Maximum elevation, $\theta_{el}^{max} = 77.9^\circ, \theta(t^{az}) = 180^\circ\}$ at $\sim 13 : 20$
- Daylight duration, $D_{day} = 14.33h$
- Azimuth range, $r_{az} = 240^\circ$
- Elevation range, $r_{el} = 2 * \theta_{el}^{max} = 155.8^\circ$

Average speed of azimuth displacement, \bar{v}_{az} , and elevation, \bar{v}_{el} , are calculated.

$$\bar{v}_{az} = \frac{r_{az}}{D_{day}} = \frac{240^\circ}{873min} = 0.27^\circ/min \quad (3.17)$$

and

$$\bar{v}_{el} = \frac{r_{el}}{D_{day}} = \frac{77.9^\circ * 2}{873min} = 0.18^\circ/min \quad (3.18)$$

Sun's speed is not constant during time as seen in 3.2, therefore worst cases(maximum speed) should be also considered for both axes. From 3.2 is observed that azimuthal worst case appears around noon while reaching maximum elevation and elevation worst case is at sunrise and sunset. Also a reasonable and representing interval, $\Delta t = t_2 - t_1 = 20min$ was selected, where t^1, t^2 are time points that meet our worst cases observations.

azimuth

$$t_1^{az} = 13 : 10 \Rightarrow \theta(t_1^{az}) = 163.1^\circ$$

$$t_2^{az} = 13 : 30 \Rightarrow \theta(t_2^{az}) = 184.6^\circ$$

,where $\theta(t^{az})$ is the specific azimuth position at time t_{az} .

So azimuth max speed, v_{az}^{max} , equals:

$$v_{az}^{max} = \frac{\theta(t_2^{az}) - \theta(t_1^{az})}{t_2^{az} - t_1^{az}} = \frac{21.5^\circ}{20min} = 1.08^\circ/min \quad (3.19)$$

elevation

The same applies in elevation also with one addition; because maximum elevation speed occurs both in sunrise and sunset their average angular displacement, $avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el}))$, is used as nominator.

$$\underbrace{\left\{ \begin{array}{l} t_1^{el} = 6 : 14 \Rightarrow \theta(t_1^{el}) = 0^\circ \\ t_2^{el} = 6 : 34 \Rightarrow \theta(t_2^{el}) = 3.6^\circ \end{array} \right. \quad \text{set} \left\{ \begin{array}{l} t_1'^{el} = 20 : 18 \Rightarrow \theta(t_1'^{el}) = 3.6^\circ \\ t_2'^{el} = 20 : 38 \Rightarrow \theta(t_2'^{el}) = 0^\circ \end{array} \right. \right. \quad (3.20)$$

$$avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el})) = \frac{(\theta(t_2^{el}) - \theta(t_1^{el})) + (\theta(t_1'^{el}) - \theta(t_2'^{el}))}{2} = 3.6^\circ \quad (3.21)$$

Hence, elevation max speed, v_{el}^{max} , equals:

$$v_{el}^{max} = \frac{avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el}))}{\Delta t^{el}} = \frac{3.6^\circ}{20min} = 0.18^\circ/min \quad (3.22)$$

Note: at 21 June $v_{el}^{max} = \bar{v}_{el}$

20 March

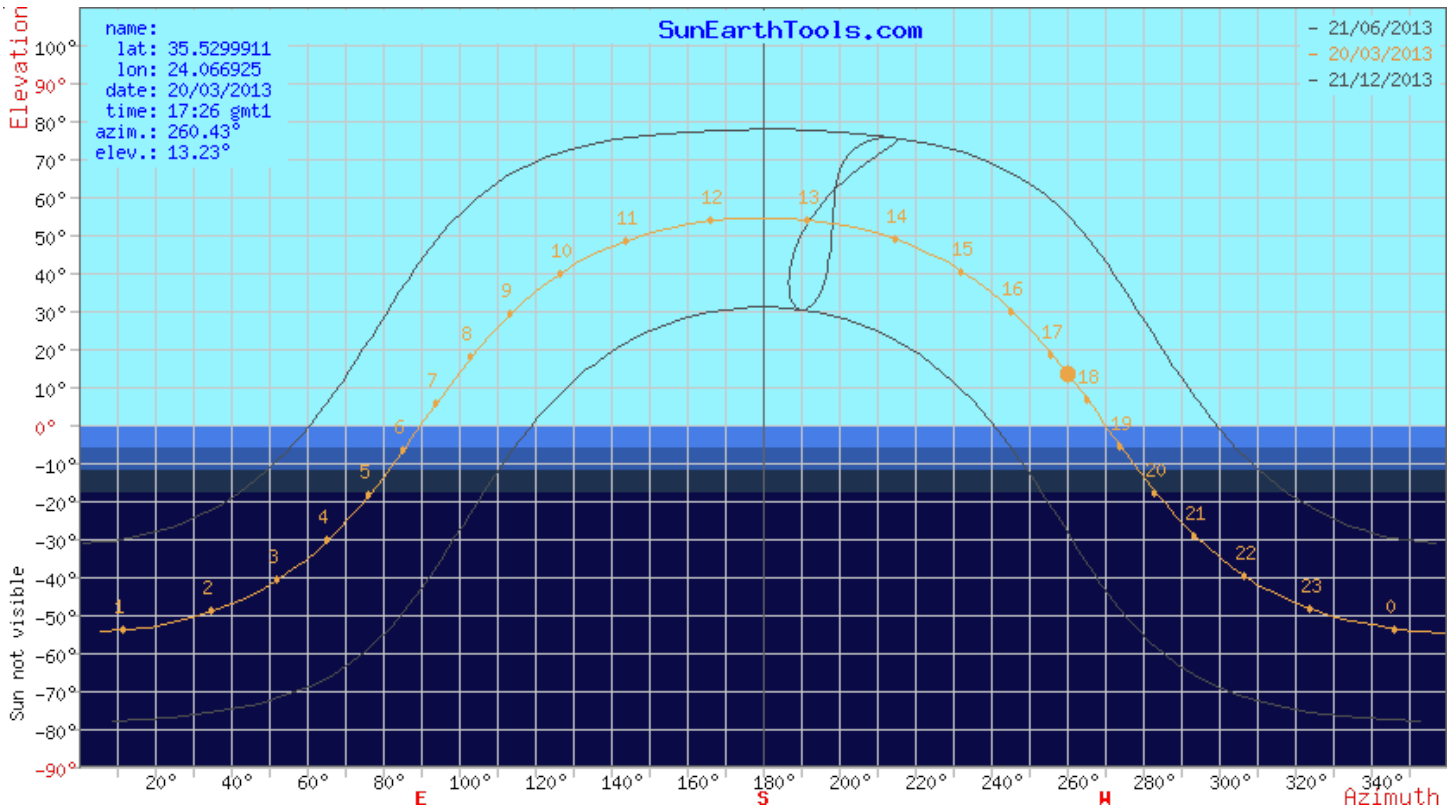


FIGURE 3.3: Cartesian-like sun route for 20 March

- Rise: $\{t_{rise} = 6 : 28 \Rightarrow \theta(t_{rise}^{az}) = 89.5^\circ, \theta(t_{rise}^{el}) = 0^\circ\}$
- Set: $\{t_{set} = 18 : 36 \Rightarrow \theta(t_{set}^{az}) = 270.7^\circ, \theta(t_{set}^{el}) = 0^\circ\}$
- Maximum elevation, $\theta_{el}^{max} = 54.5^\circ, \theta(t^{az}) = 180^\circ$ at $\sim 12 : 30$
- Daylight duration, $D_{day} = 12.08h$
- Azimuth range, $r_{az} = 181.2^\circ$

- Elevation range, $r_{el} = 2 * \theta_{el}^{max} = 109^\circ$

Average speed of azimuth displacement, \bar{v}_{az} , and elevation, \bar{v}_{el} , are calculated.

$$\bar{v}_{az} = \frac{r_{az}}{D_{day}} = \frac{181.2^\circ}{728min} = 0.25^\circ/min \quad (3.23)$$

and

$$\bar{v}_{el} = \frac{r_{el}}{D_{day}} = \frac{109^\circ * 2}{728min} = 0.15^\circ/min \quad (3.24)$$

Sun's speed is not constant during time as seen in 3.3, therefore worst cases(maximum speed) should be also considered for both axes. From 3.3 is observed that azimuthal worst case appears around noon while reaching maximum elevation and elevation worst case is at sunrise and sunset. Also a reasonable and representing interval, $\Delta t = t_2 - t_1 = 20min$ was selected, where t^1, t^2 are time points that meet our worst cases observations.

azimuth

$$t_1^{az} = 12 : 20 \Rightarrow \theta(t_1^{az}) = 175.1^\circ$$

$$t_2^{az} = 12 : 40 \Rightarrow \theta(t_2^{az}) = 183.7^\circ$$

,where $\theta(t^{az})$ is the specific azimuth position at time t_{az} .

So azimuth max speed, v_{az}^{max} , equals:

$$v_{az}^{max} = \frac{\theta(t_2^{az}) - \theta(t_1^{az})}{t_2^{az} - t_1^{az}} = \frac{8.6^\circ}{20min} = 0.43^\circ/min \quad (3.25)$$

elevation

The same applies in elevation also with one addition; because maximum elevation speed occurs both in sunrise and sunset their average angular displacement, $avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el}))$, is used as nominator.

$$\underbrace{\begin{matrix} rise & \left\{ \begin{matrix} t_1^{el} = 6 : 32 \Rightarrow \theta(t_1^{el}) = 0^\circ \\ t_2^{el} = 6 : 52 \Rightarrow \theta(t_2^{el}) = 4^\circ \end{matrix} \right. & set & \left\{ \begin{matrix} t_1'^{el} = 18 : 12 \Rightarrow \theta(t_1'^{el}) = 4^\circ \\ t_2'^{el} = 18 : 32 \Rightarrow \theta(t_2'^{el}) = 0^\circ \end{matrix} \right. \end{matrix}} \quad (3.26)$$

$$avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el})) = \frac{(\theta(t_2^{el}) - \theta(t_1^{el})) + (\theta(t_1'^{el}) - \theta(t_2'^{el}))}{2} = 4^\circ \quad (3.27)$$

Hence, elevation max speed, v_{el}^{max} , equals:

$$v_{el}^{max} = \frac{avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el}))}{\Delta t^{el}} = \frac{4^\circ}{20min} = 0.2^\circ/min \quad (3.28)$$

22 September

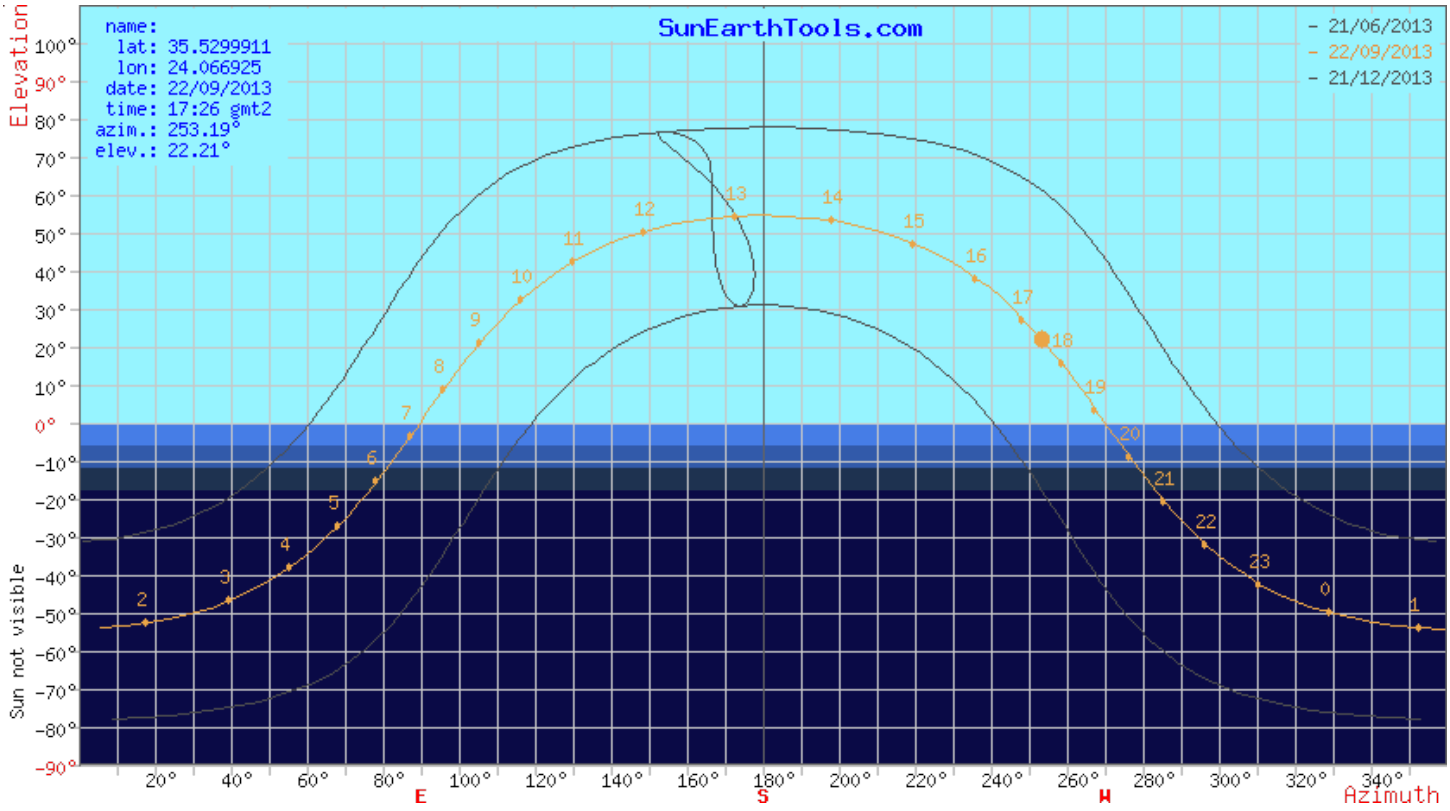


FIGURE 3.4: Cartesian-like sun route for 22 September

- Rise: $\{t_{rise} = 7 : 12 \Rightarrow \theta(t_{rise}^{az}) = 89.1^\circ, \theta(t_{rise}^{el}) = 0^\circ\}$
- Set: $\{t_{set} = 19 : 21 \Rightarrow \theta(t_{set}^{az}) = 270.7^\circ, \theta(t_{set}^{el}) = 0^\circ\}$
- Maximum elevation, $\theta_{el}^{max} = 54.7^\circ, \theta(t^{az}) = 180^\circ\}$ at $\sim 13 : 15$
- Daylight duration, $D_{day} = 12.09h$
- Azimuth range, $r_{az} = 181.6^\circ$
- Elevation range, $r_{el} = 2 * \theta_{el}^{max} = 109.4^\circ$

Average speed of azimuth displacement, \bar{v}_{az} , and elevation, \bar{v}_{el} , are calculated.

$$\bar{v}_{az} = \frac{r_{az}}{D_{day}} = \frac{181.6^\circ}{729min} = 0.25^\circ/min \quad (3.29)$$

and

$$\bar{v}_{el} = \frac{r_{el}}{D_{day}} = \frac{109.4^\circ * 2}{729min} = 0.15^\circ/min \quad (3.30)$$

Sun's speed is not constant during time as seen in 3.4, therefore worst cases(maximum speed) should be also considered for both axes. From 3.4 is observed that azimuthal worst case appears around noon while reaching maximum elevation and elevation worst case is at sunrise and sunset. Also a reasonable and representing interval, $\Delta t = t_2 - t_1 = 20min$ was selected, where t^1, t^2 are time points that meet our worst cases observations.

azimuth

$$t_1^{az} = 13 : 05 \Rightarrow \theta(t_1^{az}) = 175^\circ$$

$$t_2^{az} = 13 : 25 \Rightarrow \theta(t_2^{az}) = 183.6^\circ$$

,where $\theta(t^{az})$ is the specific azimuth position at time t_{az} .

So azimuth max speed, v_{az}^{max} , equals:

$$v_{az}^{max} = \frac{\theta(t_2^{az}) - \theta(t_1^{az})}{t_2^{az} - t_1^{az}} = \frac{8.6^\circ}{20min} = 0.43^\circ/min \quad (3.31)$$

elevation

The same applies in elevation also with one addition; because maximum elevation speed occurs both in sunrise and sunset their average angular displacement, $avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el}))$, is used as nominator.

$$\underbrace{\left. \begin{array}{l} \text{rise} \left\{ \begin{array}{l} t_1^{el} = 7 : 16 \Rightarrow \theta(t_1^{el}) = 0^\circ \\ t_2^{el} = 7 : 36 \Rightarrow \theta(t_2^{el}) = 4.1^\circ \end{array} \right. \\ \text{set} \left\{ \begin{array}{l} t_1'^{el} = 18 : 56 \Rightarrow \theta(t_1'^{el}) = 4.1^\circ \\ t_2'^{el} = 19 : 16 \Rightarrow \theta(t_2'^{el}) = 0^\circ \end{array} \right. \end{array} \right\}} \quad (3.32)$$

$$avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el})) = \frac{(\theta(t_2^{el}) - \theta(t_1^{el})) + (\theta(t_1'^{el}) - \theta(t_2'^{el}))}{2} = 4.1^\circ \quad (3.33)$$

Hence, elevation max speed, v_{el}^{max} , equals:

$$v_{el}^{max} = \frac{avg(\Delta\theta(t^{el}), \Delta\theta(t'^{el}))}{\Delta t^{el}} = \frac{4.1^\circ}{20min} = 0.21^\circ/min \quad (3.34)$$

3.2.3 Dataset construction

Based on the observations from Tables 3.4, 3.5, 3.7, 3.6 we suggest that the following dates sufficiently cover representative weather scenarios for Chania and are chosen to be the basic datasets:

1. 21 December 2008
2. 21 June 2012
3. 22 September 2012
4. 20 March 2011

subsequently weather conditions history was retrieved, from a free online source ³, and appropriately formatted. This particular weather database provides users with an API⁴ making information retrieval easier and more efficient. API requests are made over HTTP in this form: `http://api.wunderground.com/api/some_key/condition/q/location.json` and data features return JSON⁵ or XML⁶ files. In this case JSON was used and an example of this file style can be seen in fig 3.1.

```
1 { "menu": {  
2   "id": "file",  
3   "value": "File",  
4   "popup": {  
5     "menuitem": [  
6       { "value": "New", "onclick": "CreateNewDoc()" },  
7       { "value": "Open", "onclick": "OpenDoc()" },  
8       { "value": "Close", "onclick": "CloseDoc()" }  
9     ]  
10  }  
11 }}
```

LISTING 3.1: JSON example

³www.wunderground.com

⁴<http://www.wunderground.com/weather/api/>

⁵<http://www.json.org/>

⁶<http://www.w3.org/XML/>

The historical weather data acquired for Chania were recorded from a weather station located in Souda, Chania,GR⁷.

At first the returned file was stripped down to the minimum necessary information. This step was taken because large amount of irrelevant data, as to this work, was retrieved. The data fields of interest are:

1. UTC UNIX timestamp⁸
2. Temperature(C°)
3. Relative humidity(%)
4. Sky condition (fig 3.3)
5. Wind speed(km/h)
6. Wind direction($^\circ$).

Then "fill in the blanks" methodology was applied. The retrieved data were recorded with 30 minutes intervals and when compared to $\Delta_{intrv} = 5min$ a shortage is found. For this work 12 data samples are needed each hour instead of 2 that were actually recorded. In order to overcome this issue linear interpolation was applied to substitute the missing data. Weather conditions do not dramatically change every half an hour so this workaround is legit and also accurate. In detail for linear interpolation if two points $x_0 \rightarrow f(x_0) = y_0, x_1 \rightarrow f(x_1) = y_1$ are known then for $\forall x, x \in [x_0, x_1] : f(x) = y$ is given from the equation

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \iff \quad (3.35)$$

$$y = y_0 + (y_1 - y_0) \frac{x - x_0}{x_1 - x_0} \quad (3.36)$$

The previous eq 3.36 was adjusted to calculate the desired data fields, where

- x_0, x_1 are the timestamps of consecutive recorded data
- y_0, y_1 are the values for x_0, x_1 respectively
- $x \in [x_0 + \Delta_{intrv}, x_0 + 2\Delta_{intrv}, \dots, x_1 - \Delta_{intrv}]$.

After constructing the initial datasets for evaluation further experimental setups were produced. Another 4 identical to the previous ones but with constant wind speed of

⁷<http://www.wunderground.com/weather-forecast/GR/Souda.html>

⁸http://en.wikipedia.org/wiki/Unix_time

60 km/h which is a typical maximum at which PVS remains operational [48] This was done to include extreme consumption conditions. Moreover the following datasets were constructed based on artificial scenarios of our design:

1. transition from clear sky(CLR) to overcast(OVC), based on 20 March 2011 data
2. transition from overcast(OVC) to clear sky, based on 22 September 2012 data
3. complete overcast(OVC) during the whole day, based on 21 December 2008 data
4. transition from few clouds(FEW) to broken(BKN) again to few(FEW) and back to broken(BKN) at last for two cases:
 - (a) spring day, basen on 20 March 2011 data
 - (b) winter day, based on 21 December 2008 data
5. transition from broken(BKN) clouds to few(FEW) again to broken(BKN) and back to few(FEW) at last for two cases:
 - (a) spring day, basen on 20 March 2011 data
 - (b) winter day, based on 21 December 2008 data

The final collection of datasets aims to be a wide variety of representing weather conditions for Chania in order the results of this work to be generalizable.

All the previous work regarding datasets construction was done with a Python⁹ script written for the occasion.

3.3 Moving windows method

3.3.1 Design and properties

As seen in eq 3.5 in page 21 and using contents from Table 3.2 in page 20 our problem's state space is

$$151 \times 36 \times 118 \leq |S| \leq 151 \times 36 \times 175 \quad (3.37)$$

$$641.448 \leq |S| \leq 951.300 \quad (3.38)$$

⁹<https://www.python.org/>

and the action space from eq 3.6 in page 21, with permitted actions corresponding to the transitions for all orientation positions

$$|A| = 151 \times 36 \quad (3.39)$$

$$|A| = 5.436 \quad (3.40)$$

This problem space, $|S \times A|$, is rather large and any solution method would be intractable to achieve optimal policy. In order to resize the problem scale we introduce an approach, mentioned from now on as *WindowVI*, that reduces significantly the action space A . Specifically, from current state $s_c \in S$, in timestep τ , the accessible states, under a set of actions A_{s_c} , are included in a rectangular frame around current state s_c with both dimensions corresponding to azimuth, κ_{s_c} , and elevation, λ_{s_c} positions. This can be considered as a surrounding window of the current state, inside of which all states are accessible following the corresponding action. As a result $\forall s \in S_{s_c}, \alpha_{s_c} : (\kappa_{s_c}, \lambda_{s_c})$, where

- $\kappa_{s_c} \in [1, |K_{s_c}|]$ with $|K_{s_c}| < |K|$
- $\lambda_{s_c} \in [1, |\Lambda_{s_c}|]$ with $|\Lambda_{s_c}| < |\Lambda|$.

For determining $|K_{s_c}|, |\Lambda_{s_c}|$ further analysis is required. This analysis is based on the astronomical results produced in subsection 3.2.2 in page 25 because the proposed configurations must be able to orient the system according to sun's movement, at least, for both axes. The minimum dimensions for this to happen are

- **azimuth movement**

sun's maximum azimuth speed is observed in 21 June, eq 3.19 page 28, thus

$$v_{az}^{max} * \Delta_{intrv} = 1.08^\circ/min * 5min = 5.4^\circ \quad (3.41)$$

- **elevation movement**

sun's maximum elevation speed is observed in 22 September, eq 3.34 page 32, thus

$$v_{el}^{max} * \Delta_{intrv} = 0.21^\circ/min * 5min = 1.05^\circ \quad (3.42)$$

From the equation 3.41 and the system's angular displacement, θ_{step} , $|K_{s_c}|$ can be approached as

$$|K_{s_c}| \geq \left\lceil \frac{5.4^\circ}{\theta_{step}} \right\rceil + 1 \quad (3.43)$$

$$|K_{s_c}| \geq \left\lceil \frac{5.4^\circ}{1.8^\circ} \right\rceil + 1 \quad (3.44)$$

$$|K_{s_c}| \geq 4 \quad (3.45)$$

one added position is needed for s_c current position. Similarly, for $|\Lambda_{s_c}|$

$$|\Lambda_{s_c}| \geq 2 \left\lceil \frac{1.05^\circ}{\theta_{step}} \right\rceil + 1 \quad (3.46)$$

$$|\Lambda_{s_c}| \geq 2 \left\lceil \frac{1.05^\circ}{1.8^\circ} \right\rceil + 1 \quad (3.47)$$

$$|\Lambda_{s_c}| \geq 3 \quad (3.48)$$

double plus one positions are needed for up/down movement.

The resulted dimensions are sufficient to positioning continuously the system direct to sun's route. As stated in [49–51] sky conditions and solar radiation components are strongly correlated, meaning that the source of solar radiation varies in the sky dome, therefore bigger dimensions were adopted to improve efficiency of the method. The following configurations are proposed

- $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{80 \text{ positions}}$

- $|K_{s_c}| = 4, |\Lambda_{s_c}| = 20$

- $|K_{s_c}| = 5, |\Lambda_{s_c}| = 16$

- $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{60 \text{ positions}}$

- $|K_{s_c}| = 3, |\Lambda_{s_c}| = 20$

- $|K_{s_c}| = 4, |\Lambda_{s_c}| = 15$

- $|K_{s_c}| = 5, |\Lambda_{s_c}| = 12$

- $|K_{s_c}| = 6, |\Lambda_{s_c}| = 10$

- $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{45 \text{ positions}}$

- $|K_{s_c}| = 3, |\Lambda_{s_c}| = 15$

- $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{40 \text{ positions}}$

- $|K_{s_c}| = 4, |\Lambda_{s_c}| = 10$

- $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{20 \text{ positions}}$
 - $|K_{s_c}| = 4, |\Lambda_{s_c}| = 5$

For each case, for current state $s_c : (\kappa_{s_c}, \lambda_{s_c}, \tau_{s_c})$, $\kappa_{s_c} = 1$, except when $|K_{s_c}| = 5$, then $\kappa_{s_c} = 2$. In addition, for every case $\lambda_{s_c} = \left\lfloor \frac{|\Lambda_{s_c}|}{2} + 1 \right\rfloor$. λ_{s_c} is positioned in the middle of vertical dimension for ascending and descending operations of the system.

Special cases for $\kappa_{s_c}, \lambda_{s_c}$ As seen in Table 3.2 page 20 $|K| = 151$ and $|\Lambda| = 36$, hence functions were developed for when limits are met in azimuth and elevation axis. The target is to keep the selected action space size by repositioning it appropriately. As s_c is relevant to surrounding frame of positions and s is absolutely positioned inside the complete space of positions, let current state $s_c : (\kappa_{s_c}, \lambda_{s_c}, \tau_{s_c})$ be in reference within the $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}|$ frame and state $s : (\kappa_s, \lambda_s, \tau_s)$ be in reference within $|A_s| = |K_s \times \Lambda_s|$, with $s_c \triangleq s$.

Azimuth cases

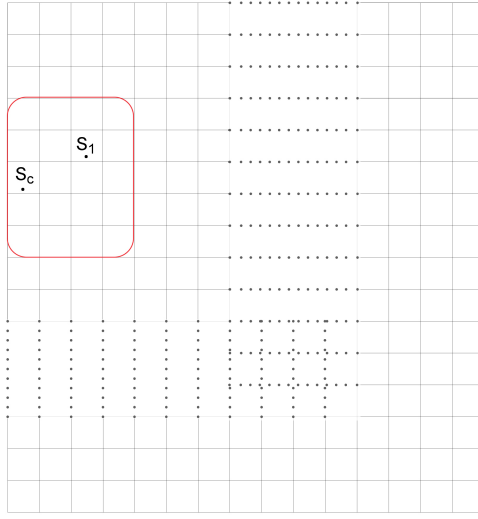
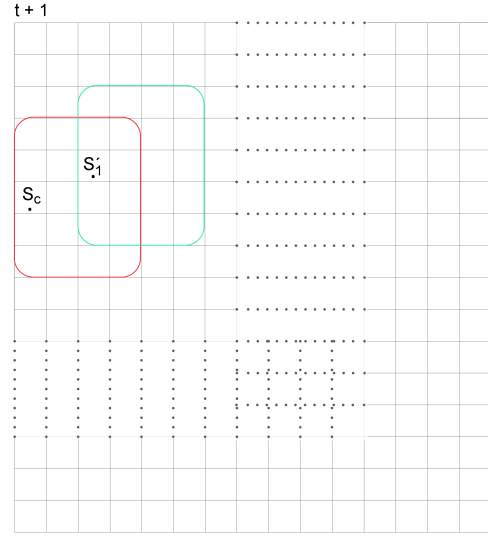
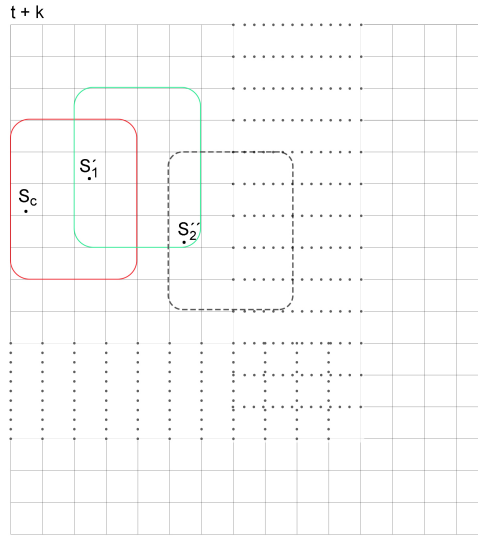
- If current state s_c is close enough to first position in azimuth dimension then action space is shifted by $\kappa_{s_c} - \kappa_s$ positions. If $\kappa_{s_c} > \kappa_s$, then $\kappa'_{s_c} = \kappa_s$.
- current state s_c is close enough to last position in azimuth dimension then action space is shifted by $\kappa_s - \kappa_{s_c}$ positions. If $\kappa_s - \kappa_{s_c} + |K_{s_c}| > |K_s|$, then $\kappa'_{s_c} = \kappa_s + |K_{s_c}| - |K_s|$

Elevation cases

- When current state s_c is close enough to first position in elevation dimension then action space is shifted by $\lambda_{s_c} - \lambda_s$ positions. If $\lambda_{s_c} > \lambda_s$, then $\lambda'_{s_c} = \lambda_s$.
- When current state s_c is close enough to last position in elevation dimension then action space is shifted by $\lambda_s - \lambda_{s_c}$ positions. If $\lambda_s - \lambda_{s_c} + |\Lambda_{s_c}| > |\Lambda_s|$, then $\lambda'_{s_c} = \lambda_s + |\Lambda_{s_c}| - |\Lambda_s|$

3.3.2 k -step look ahead

Furthermore a look ahead functionality was implemented in order to provide some insight for the energy production of accessible positions in future timesteps. The intention is future information regarding energy gain per reachable state to be back propagated into

FIGURE 3.5: Available states from s_c at t FIGURE 3.6: Available states from s'_1 at $t+1$ FIGURE 3.7: Available states from s''_2 at $t+k$

current timestep. This way control decisions made in every timestep would have taken into account potential situations. In more detail, let $s_c : (\kappa_{s_c}, \lambda_{s_c}, \tau_{s_c})$ be the current state in timestep t and S_{s_c} be the set of accessible states from s_c under action set A_{s_c} within $|K_{s_c} \times \Lambda_{s_c}|$ positions, as denoted by the red window surrounding s_c in figure 3.5. For each state inside this frame we calculate its corresponding value. This value consists of the energy gain in this state plus the maximum value of accessible states in future timesteps. In figure 3.6 we describe the available states from state s'_1 in timestep $t+1$ and subsequently the accessible states from s''_2 , in timestep $t+k$ generally, with a dashed frame in figure 3.7. In this example the value of state s_1 in timestep t depends

on the maximum value of the states available in $t + 1$ and so on until timestep $t + k$ is reached. This procedure is iteratively continued for all states accessible in timestep t , figure 3.5, in order to construct their values. When value iteration is finished the state with maximum value is chosen to be the next current state of the system, s_c .

Because VI complexity grows as lookahead steps increase, the following configuration was chosen for the moving windows in order the algorithm to be feasible.

- $k = 1$
 - $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{60 \text{ positions}}$
 - * $|K_{s_c}| = 3, |\Lambda_{s_c}| = 20$
 - * $|K_{s_c}| = 4, |\Lambda_{s_c}| = 15$
 - * $|K_{s_c}| = 5, |\Lambda_{s_c}| = 12$
 - * $|K_{s_c}| = 6, |\Lambda_{s_c}| = 10$
 - $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{45 \text{ positions}}$
 - * $|K_{s_c}| = 3, |\Lambda_{s_c}| = 15$
 - $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{40 \text{ positions}}$
 - * $|K_{s_c}| = 4, |\Lambda_{s_c}| = 10$
 - $|A_{s_c}| = |K_{s_c} \times \Lambda_{s_c}| = \mathbf{20 \text{ positions}}$
 - * $|K_{s_c}| = 4, |\Lambda_{s_c}| = 5$
- $k = 2$
 - $|K_{s_c}| = 4, |\Lambda_{s_c}| = 5$
- $k = 3$
 - $|K_{s_c}| = 3, |\Lambda_{s_c}| = 3$

The main algorithm at first chooses greedily the initial current state by simply searching for the max reward at $\tau_s = 0$. Then it constructs S_{s_c} , the surrounding window, given the s_c . For all states accessible from s_c state values are calculated through the lookahead process, as described above. When the value iteration is finished, the value functions for these states have been calculated. The state with maximum value is then picked as the current state for timestep $\tau_s + 1$. This procedure is repeated until the last timestep is reached. The set of all the states that were picked through this process, for all timesteps, is stored in a vector and represents the PVS orientation positions for the day. An algorithmic scheme is presented in algorithm 3.

Algorithm 3 WindowVI

procedure WINDOWVI-choose initial s_c ▷ first current state**for all** $\tau_s \in I$ **do**- calculate S_{s_c} based on s_c ▷ S_{s_c} : set of available states**for all** $s \in S_{s_c}$ **do****for** $i = 0$ to k , $i++$ **do** ▷ k : horizon steps

$$V_i(s) \leftarrow R(s, \alpha) + \max_a \sum_{s'} P_\alpha(s, s') V_{i-1}(s')$$

$$\pi(s, i) \leftarrow \arg \max_\alpha V_i(s)$$

end for**end for** $s_c \leftarrow \arg \max_s V(s)$ ▷ choose maximum value state as next state $route[\tau_s] \leftarrow s_c$ ▷ store all chosen states in a vector**end for****end procedure**

Chapter 4

Simulation results

In this chapter simulation results are presented. All configurations analysed in section 3.3 in page 35 for every dataset built, as presented in section 3.2.3 in page 33, are compared against a typical 2-axis sun tracker which follows an astronomical function calculating the sun's position at any given moment.

In every table value(s) in bold correspond to maximum for each case.

- 20 March 2011

Sky conditions historical data for this day include scattered clouds from sunrise until noon. Then cloud cover increases to broken clouds and complete overcast after that, to end up in broken clouds again before sunset, as seen in Table 3.7 in page 24.

TABLE 4.1: Results for 20 March 2011

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			32.021
0-step	80		
		4×20	33.018
		5×16	33.028
	60		
		3×20	33.018
		4×15	33.028
		5×12	33.030
		6×10	33.031
	45		

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
1-step	40	3×15	33.028
		4×10	33.031
	20	4×5	33.008
		6×10	33.034
1-step	45	3×20	33.034
		4×15	33.033
	40	5×12	33.027
		6×10	33.034
2-step	40	3×15	33.034
		4×10	33.034
	20	4×5	33.022
		6×10	33.034
3-step	20	4×5	33.022
		6×10	33.034
	9	3×3	32.938
		4×4	33.034

For 20 March 2011, $|K_{sc} \times \Lambda_{sc}| = i.) 3 \times 20$, $ii.) 3 \times 15$, $iii.) 4 \times 10$ and $iv.) 6 \times 10$, with 1-step look ahead option, exhibit better performance than the other configurations. Also the rest window size configurations perform better with 1-step lookahead functionality except for 5×12 case which shows a slight drop. The best performing configurations, interestingly, are consisting of 3 azimuth positions or 10 elevation positions. Clearly every setup outperforms the typical 2-axis tracker.

- 22 September 2012

During this day archival data include mild cloud cover conditions. Scattered clouds at sunrise are followed from few clouds for the rest of the day just before sunset when the sky is totally clear as seen in Table 3.6 in page 24.

TABLE 4.2: Results for 22 September 2012

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			51.624
0-step			
	80		
		4×20	52.304
		5×16	52.305
	60		
		3×20	52.305
		4×15	52.305
		5×12	52.312
		6×10	52.312
	45		
		3×15	52.306
	40		
		4×10	52.312
	20		
		4×5	52.305
1-step			
	60		
		3×20	52.290
		4×15	52.305
		5×12	52.310
		6×10	52.320
	45		
		3×15	52.306
	40		
		4×10	52.321
	20		
		4×5	52.316
2-step			
	20		
		4×5	52.316
3-step			
	9		
		3×3	52.228

For 22 September 2012, $|K_{s_c} \times \Lambda_{s_c}| = 4 \times 10$, with 1-step look ahead option, yields marginally better results than the second best configuration, $|K_{s_c} \times \Lambda_{s_c}| = 6 \times 10$, with 1-step look ahead option. Setup with maximum available elevation positions, 3×20 , along with setup 5×12 , with 1-step lookahead for both, present a performance drop whereas setups with less elevation positions (6×10 , 4×10 , 4×5) increase their performance. Two setups of identical elevation positions (4×15 , 3×15) remain in same levels for both 0-step and 1-step lookahead functionality. Also, every setup outperforms the typical 2-axis tracker.

- 21 June 2012

For a summer day in Greece the sky conditions are the ones anticipated. Clear sky is observed through the whole day as seen in Table 3.5 in page 24.

TABLE 4.3: Results for 21 June 2012

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			73.434
0-step	80		
		4×20	73.981
		5×16	73.990
	60	3×20	73.956
		4×15	73.983
		5×12	73.988
		6×10	73.994
	45		
		3×15	73.959
	40		
		4×10	73.994
	20		
		4×5	73.995
1-step	60	3×20	73.960
		4×15	73.991
		5×12	73.987
		6×10	73.997
	45		

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
		3×15	73.966
	40		
		4×10	73.997
	20		
		4×5	73.995
2-step			
	20		
		4×5	73.995
3-step			
	9		
		3×3	73.922

For 21 June 2012, $|K_{s_c} \times \Lambda_{s_c}| = i.) 4 \times 10$ and $ii.) 6 \times 10$, with 1-step lookahead option, are the top performing configurations closely to the next ones. Setups with minimum available azimuth positions, such 3×3 , 3×15 and 3×20 , and regardless lookahead step functionality present reduced performance compared to the rest ones. This is something expected though. During a clear sunny summer day, the nearly optimal strategy is generally similar to the typical sun tracker's, meaning to stay aligned with sun's movement, in order to absorb the beam component of total solar radiation, as described in section 2.2.1 in page 5. Sun's azimuth movement speed is $1.08^\circ/min$, from eq. 3.19, and angular displacement after $\Delta_{intrv} = 5min$ is 5.4° . With two available positions for azimuth movement(one is already occupied as the *current* position) only 3.6° of angular displacement can be achieved, resulting in a hysteresis between system-sun alignment. Every configuration shows an improvement with 1-step lookahead over 0-step lookahead apart from 5×12 setup. Interestingly every setup outperforms the typical 2-axis tracker even though the sky conditions are completely clear, probably because of the improved performance during sunrise and sunset, when the main amount of solar radiation is not included in the beam component.

- 21 December 2008

On this winter day the sky is heavily covered with clouds. At sunrise an overcast is observed followed by broken clouds for the rest of the day, as seen in Table 3.4 in page 23.

TABLE 4.4: Results for 21 December 2008

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
----------------------	-------------	----------------------------------	---------------------

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
solar tracker			11.465
0-step			
	80		
		4×20	11.763
		5×16	11.769
	60		
		3×20	11.762
		4×15	11.770
		5×12	11.778
		6×10	11.771
	45		
		3×15	11.770
	40		
		4×10	11.771
	20		
		4×5	11.767
1-step			
	60		
		3×20	11.751
		4×15	11.764
		5×12	11.776
		6×10	11.780
	45		
		3×15	11.764
	40		
		4×10	11.779
	20		
		4×5	11.782
2-step			
	20		
		4×5	11.782
3-step			
	9		
		3×3	11.760

For 21 December 2008, $|K_{sc} \times \Lambda_{sc}| = 4 \times 5$ configuration, with 1-step and 2-step lookahead option respectively, outperform the other ones. This setup shows a

significant improvement over 0-step lookahead. The only setups that present a performance increase with 1-step lookahead option are the 4×10 , 6×10 ones, with both having ten elevation positions. Additionally every setup is better than the typical 2-axis tracker.

- **20 March 2011 with transition from clear sky(CLR) to overcast(OVC).**

In this scenario sky conditions begin from clear and gradually convert to overcast passing from all intermediate cloud cover levels. This transition happens over equally distributed time intervals from sunrise until sunset.

TABLE 4.5: Results for 20 March 2011, scenario [CLR-OVC]

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			42.589
0-step			
	80		
		4×20	43.295
		5×16	43.300
	60		
		3×20	43.295
		4×15	43.301
		5×12	43.305
		6×10	43.309
	45		
		3×15	43.301
	40		
		4×10	43.309
	20		
		4×5	43.301
1-step			
	60		
		3×20	43.297
		4×15	43.305
		5×12	43.305
		6×10	43.311
	45		
		3×15	43.304
	40		
		4×10	43.311

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
	20		
		4×5	43.306
2-step			
	20		
		4×5	43.306
3-step			
	9		
		3×3	43.258

For 20 March 2011 with transition from clear sky(CLR) to overcast(OVC), $|K_{s_c} \times \Lambda_{s_c}| = i.) 4 \times 10$ and $ii.) 6 \times 10$, with 1-step lookahead option, present the top results among the tested setups. Similar to 20 March 2011 results, Table 4.1, all configurations exhibit better performance, with 1-step lookahead in this case, even 5×12 setup that presented a marginal decrease in the first now remains at the same levels of performance for 0 and 1-step lookahead respectively. Moreover every configuration is better than the typical 2-axis tracker.

- **22 September 2012 with transition from overcast(OVC) to clear sky(CLR).**

In this artificial scenario sky conditions begin from complete overcast and gradually end up in clear passing from all intermediate cloud cover levels. This transition happens over equally distributed time intervals from sunrise until sunset.

TABLE 4.6: Results for 22 September 2012, scenario [OVC-CLR]

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			43.890
0-step			
	80		
		4×20	44.746
		5×16	44.746
	60		
		3×20	44.746
		4×15	44.747
		5×12	44.750
		6×10	44.759
	45		
		3×15	44.747
	40		

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
1-step	20	4×10	44.759
		4×5	44.778
		3×20	44.748
		4×15	44.756
1-step	60	5×12	44.764
		6×10	44.770
		3×15	44.760
		4×10	44.778
2-step	45	4×5	44.776
		4×5	44.776
		3×3	44.692
		4×5	44.776
3-step	20	4×5	44.776
		4×5	44.776
		3×3	44.692
		4×5	44.776
3-step	9	4×5	44.776
		4×5	44.776
		3×3	44.692
		4×5	44.776

For 22 September 2012 with transition from overcast(OVC) to clear sky(CLR), $|K_{sc} \times \Lambda_{sc}| = i.) 4 \times 5$ and $ii.) 4 \times 10$, with 0-step lookahead functionality and 1-step lookahead respectively, present the best results. This is the first time that a top result comes from a configuration without lookahead functionality. These two setups have smaller set of available positions than the rest. Almost every other setup increases performance with 1-step lookahead compared to 0-step lookahead. The only setup with reduced results, even marginally, is 4×5 configuration with 1-step and 2-step lookahead options. Obviously every configuration is better than the typical 2-axis tracker.

- **20 March 2011 with transition from few clouds(FEW) to broken(BKN) to few(FEW) again and broken at last(BKN).**

In this case sky conditions are successively altered from light cloud cover to heavy, two times during the day passing through intermediate cloud cover conditions.

TABLE 4.7: Results for 20 March 2011, scenario [FEW-BKN-FEW-BKN]

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			39.810
0-step			
	80		
		4×20	40.607
		5×16	40.615
	60		
		3×20	40.607
		4×15	40.615
		5×12	40.619
		6×10	40.620
	45		
		3×15	40.615
	40		
		4×10	40.620
	20		
		4×5	40.601
1-step			
	60		
		3×20	40.603
		4×15	40.610
		5×12	40.612
		6×10	40.621
	45		
		3×15	40.610
	40		
		4×10	40.621
	20		
		4×5	40.611
2-step			
	20		
		4×5	40.611
3-step			
	9		
		3×3	40.513

For 20 March 2011 with transition from few clouds(FEW) to broken(BKN) to few(FEW) again and broken at last(BKN), $|K_{sc} \times \Lambda_{sc}| = i.) 6 \times 10$ and $ii.) 4 \times 10$ setups, with 1-step lookahead functionality, present the best results. These two configurations outperform the rest for both 0 and 1-step lookahead functionality. Every other setup with 1-step lookahead performs worse than 0-step except 4×5 setup which shows a slight improvement. As seen every configuration is better than the typical 2-axis tracker.

- **21 December 2008 with transition from few clouds(FEW) to broken(BKN) to few(FEW) again and broken at last(BKN).**

This day is the winter equivalent, in terms of sky conditions, with the previous case.

TABLE 4.8: Results for 21 December 2008, scenario [FEW-BKN-FEW-BKN]

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
solar tracker	0-step		16.485
		80	
		4×20	16.887
		5×16	16.887
		60	
		3×20	16.888
		4×15	16.887
	1-step	5×12	16.892
		6×10	16.896
		45	
		3×15	16.888
		40	
		4×10	16.896
		20	
		4×5	16.893
	1-step	60	
		3×20	16.878
		4×15	16.888
		5×12	16.893
		6×10	16.899
		45	

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
		3×15	16.888
	40		
		4×10	16.898
	20		
		4×5	16.904
2-step			
	20		
		4×5	16.904
3-step			
	9		
		3×3	16.883

For 21 December 2008 with transition from few clouds(FEW) to broken(BKN) to few(FEW) again and broken at last(BKN), $|K_{s_c} \times \Lambda_{s_c}| = 4 \times 5$ configuration, with 1-step and 2-step look ahead option respectively, outperform the rest ones. The only setup that performs better without lookahead functionality is 3×20 in contrast with the rest that improve slightly their results. Furthermore every setup is better than the typical 2-axis tracker.

- **20 March 2011 with transition from broken clouds(BKN) to few(FEW) to broken(BKN) again and few(FEW) at last.**

In this case sky conditions are successively altered from broken clouds to few, two times during the day passing through intermediate cloud cover conditions.

TABLE 4.9: Results for 20 March 2011, scenario [BKN-FEW-BKN-FEW]

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			39.846
0-step			
	80		
		4×20	40.825
		5×16	40.826
	60		
		3×20	40.825
		4×15	40.826
		5×12	40.827
		6×10	40.827
	45		

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
1-step	40	3×15	40.826
		4×10	40.827
	20	4×5	40.806
		6×10	40.832
1-step	45	3×20	40.832
		4×15	40.823
	40	5×12	40.827
		6×10	40.832
2-step	40	3×15	40.823
		4×10	40.832
	20	4×5	40.827
		6×10	40.832
3-step	20	4×5	40.827
		6×10	40.832
	9	3×3	40.741
		4×4	40.832

For 20 March 2011 with transition from broken clouds(BKN) to few(FEW) to broken(BKN) again and few(FEW) at last, $|K_{sc} \times \Lambda_{sc}| = i.) 3 \times 20$, $ii.) 4 \times 10$ and $iii.) 6 \times 10$, with 1-step look ahead option, exhibit better results than the other configurations. Most of the setups perform better with 1-step lookahead except two cases, 3×15 and 4×15 . Both have 15 elevation positions and mutually exhibit equal performance for 0 and 1-step lookahead. Also every setup outperforms the typical 2-axis tracker.

- **21 December 2008 with transition from broken clouds(BKN) to few(FEW) to broken(BKN) again and few(FEW) at last.**

This day is the winter equivalent, in terms of sky conditions, with the previous case.

TABLE 4.10: Results for 21 December 2008, scenario [BKN-FEW-BKN-FEW]

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			16.407
0-step			
	80		
		4×20	16.845
		5×16	16.844
	60		
		3×20	16.845
		4×15	16.844
		5×12	16.847
		6×10	16.848
	45		
		3×15	16.844
	40		
		4×10	16.848
	20		
		4×5	16.837
1-step			
	60		
		3×20	16.838
		4×15	16.844
		5×12	16.844
		6×10	16.856
	45		
		3×15	16.844
	40		
		4×10	16.855
	20		
		4×5	16.858
2-step			
	20		
		4×5	16.858
3-step			
	9		
		3×3	16.833

For 21 December 2008 with transition from broken clouds(BKN) to few(FEW) to broken(BKN) again and few(FEW) at last, $|K_{s_c} \times \Lambda_{s_c}| = 4 \times 5$ configuration, with 1-step and 2-step look ahead option respectively, marginally outperform the rest of the best. The k -step look ahead functionality boosts performance sufficiently in this case. Also most of the setups present same or better performance with 1-step lookahead and only 3×20 and 5×12 configurations have decreased performance with 1-step lookahead option. In addition, every setup is better than the typical 2-axis tracker.

- **21 December 2008 with complete overcast(OVC).**

This case is self explanatory, overcast is consistent throughout the day.

TABLE 4.11: Results for 21 December 2008, scenario [OVC]

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker	0-step		6.414
		80	
		4×20	6.569
		5×16	6.577
		60	
		3×20	6.568
		4×15	6.579
		5×12	6.594
		6×10	6.581
		45	
		3×15	6.579
		40	
		4×10	6.581
		20	
1-step	60	4×5	6.576
		3×20	6.556
		4×15	6.577
		5×12	6.586
		6×10	6.591
		45	
		3×15	6.579

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
	40		
		4×10	6.593
	20		
		4×5	6.591
2-step	20		
		4×5	6.591
3-step	9		
		3×3	6.564

For 21 December 2008 with complete overcast(OVC), $|K_{sc} \times \Lambda_{sc}| = 5 \times 12$ configuration, without any look ahead option, is slightly better than the rest of the best. This is only the second case that the best performance is achieved without lookahead functionality. The effect of step lookahead in performance is rather divided here with three setups presenting a loss, one remaining at the same levels and three increasing. In addition, every setup is better than the typical 2-axis tracker.

Subsequently the results derived from the datasets with enhanced wind conditions, are demonstrated. The following datasets are based on the four main datasets presented in section 3.2.3 in page 33. Sky conditions in the following cases are the same with these datasets.

- **20 March 2011 with wind speed 60km/h**

TABLE 4.12: Results for 20 March 2011 with wind speed 60km/h

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
solar tracker			31.899
0-step	80		
		4×20	32.859
		5×16	32.876
	60		
		3×20	32.859
		4×15	32.877
		5×12	32.882

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
	45	6×10	32.884
		3×15	32.877
	40	4×10	32.884
		4×5	32.859
1-step	60	3×20	32.931
		4×15	32.916
		5×12	32.885
		6×10	32.896
	45	3×15	32.921
		4×10	32.896
	40	4×5	32.874
		4×5	32.874
2-step	20	4×5	32.874
		3×3	32.792
3-step	9	3×3	32.792

For 20 March 2011 with wind speed $60km/h$, $|K_{sc} \times \Lambda_{sc}| = 3 \times 20$ configuration, with 1-step lookahead option, is the best performing one. This specific setup is also among the best for the dataset 20 March 2011 with historical weather data. The next setup regarding performance has also 3 positions in azimuth direction(3×15). As seen all setups improve their performance with 1-step lookahead. In addition, every setup is better than the typical 2-axis tracker.

- **22 September 2012 with wind speed $60km/h$**

TABLE 4.13: Results for 22 September 2012 with wind speed $60km/h$

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
----------------------	------------	--------------------------------	---------------------

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			51.515
0-step			
	80		
		4×20	52.096
		5×16	52.095
	60		
		3×20	52.098
		4×15	52.099
		5×12	52.132
		6×10	52.131
	45		
		3×15	52.101
	40		
		4×10	52.132
	20		
		4×5	52.154
1-step			
	60		
		3×20	52.016
		4×15	52.116
		5×12	52.136
		6×10	52.159
	45		
		3×15	52.095
	40		
		4×10	52.164
	20		
		4×5	52.166
2-step			
	20		
		4×5	52.166
3-step			
	9		
		3×3	52.097

For 22 September 2012 with wind speed $60\text{km}/h$, $|K_{s_c} \times \Lambda_{s_c}| = 4 \times 5$ configuration, with 1-step and 2-step lookahead option respectively, marginally outperform the

second best, $|K_{s_c} \times \Lambda_{s_c}| = 4 \times 10$ with 1-step look ahead. The top performing configuration is better than the rest at each 0-step and 1-step lookahead case respectively. Only two setups with equal azimuth positions, 3×15 and 3×20 , present a drop in their performance from 0 to 1-step lookahead. Clearly every setup is better than the typical 2-axis tracker.

- **21 June 2012 with wind speed $60km/h$**

TABLE 4.14: Results for 21 June 2012 with wind speed $60km/h$

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
solar tracker			73.264
0-step	80		
		4×20	73.782
		5×16	73.796
	60		
		3×20	73.757
		4×15	73.784
		5×12	73.804
		6×10	73.822
	45		
		3×15	73.760
	40		
		4×10	73.822
	20		
		4×5	73.841
1-step	60		
		3×20	73.745
		4×15	73.793
		5×12	73.800
		6×10	73.823
	45		
		3×15	73.767
	40		
		4×10	73.825
	20		
		4×5	73.840

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
2-step	20		
		4×5	73.840
3-step	9		
		3×3	73.775

For 21 June 2012 with wind speed $60km/h$, $|K_{sc} \times \Lambda_{sc}| = 4 \times 5$ configuration, with 0-step lookahead option, performs better than the other tested cases. The second best configuration is with same window dimensions and added lookahead steps. The observations made for Table 4.3 apply here as well. Configurations with limited azimuth positions, 3×15 and 3×20 , exhibit a noticeably reduced performance compared to the rest regardless lookahead functionality. Moreover it can be seen that every setup is better than the typical 2-axis tracker.

- **21 December 2008 with wind speed $60km/h$**

TABLE 4.15: Results for 21 December 2008 with wind speed $60km/h$.

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production(kWh)
solar tracker			11.411
0-step	80		
		4×20	11.663
		5×16	11.690
	60		
		3×20	11.661
		4×15	11.694
		5×12	11.706
		6×10	11.695
	45		
		3×15	11.694
	40		
		4×10	11.696
	20		
		4×5	11.712
1-step	60		

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production(kWh)
		3×20	11.646
		4×15	11.693
		5×12	11.698
		6×10	11.710
	45		
		3×15	11.692
	40		
		4×10	11.710
	20		
		4×5	11.726
2-step	20		
		4×5	11.726
3-step	9		
		3×3	11.699

For 21 December 2008 with wind speed 60km/h , $|K_{s_c} \times \Lambda_{s_c}| = 4 \times 5$ configuration, with 1-step and 2-step lookahead option respectively, present the best results among the other tested cases. Finally every setup is better than the typical 2-axis tracker.

Results Comparison

It is obvious that during enhanced wind conditions the best configuration is $|K_{s_c} \times \Lambda_{s_c}| = 4 \times 5$ with 1-step and 2-step lookahead options. The small size dimensions of the previous setups works as an advantage, due to the proportional correlation between energy consumption and available actions set, A_{s_c} , under specific wind conditions; the bigger the positioning displacement is, the more energy is needed for.

In contrast with the above, where certain configurations are clearly outperforming the rest, an inconclusive picture is derived from the results tables, regarding the performance hierarchy of the selected setups for the remaining datasets. For this reason, the cumulative production gain from each dataset is summed up for each setup.

$$\sum_d [Prod_j] \quad (4.1)$$

where

d each case from all datasets

$Prod_j$ is the energy production for each one of the $|K_{s_c} \times \Lambda_{s_c}|$, k -step combinations

TABLE 4.16: Accumulated energy production results.

k -step look ahead	$ A_{s_c} $	$ K_{s_c} \times \Lambda_{s_c} $	Net production sum(kWh)
solar tracker			373.986
0-step			
	80		
		4×20	380.839
		5×16	380.886
	60		
		3×20	380.816
		4×15	380.885
		5×12	380.943
		6×10	380.947
	45		
		3×15	380.862
	40		
		4×10	380.948
	20		
		4×5	380.868
1-step			
	60		
		3×20	380.787
		4×15	380.896
		5×12	380.931
		6×10	381.012
	45		
		3×15	380.879
	40		
		4×10	381.019
	20		
		4×5	380.988
2-step			
	20		
		4×5	380.988
3-step			
	9		

k -step look ahead	$ A_{sc} $	$ K_{sc} \times \Lambda_{sc} $	Net production sum(kWh)
		3×3	380.331

From the above, two configurations are distinctively better than the rest. For $|K_{sc} \times \Lambda_{sc}| = i.) 4 \times 10$ and $ii.) 6 \times 10$, with 1-step look ahead functionality the accumulated energy production exceeds $381kWh$ whereas all the other setups do not. An interesting fact is that for both of them $|A_{sc}| = 10$. Also every setup presents better results than the typical 2-axis solar tracker.

k-step lookahead behaviour

Some configurations perform better with 0-step lookahead and others with 1-step. In order to have a qualitative look over this observation, a plot with the accumulated production against each window configuration was created.

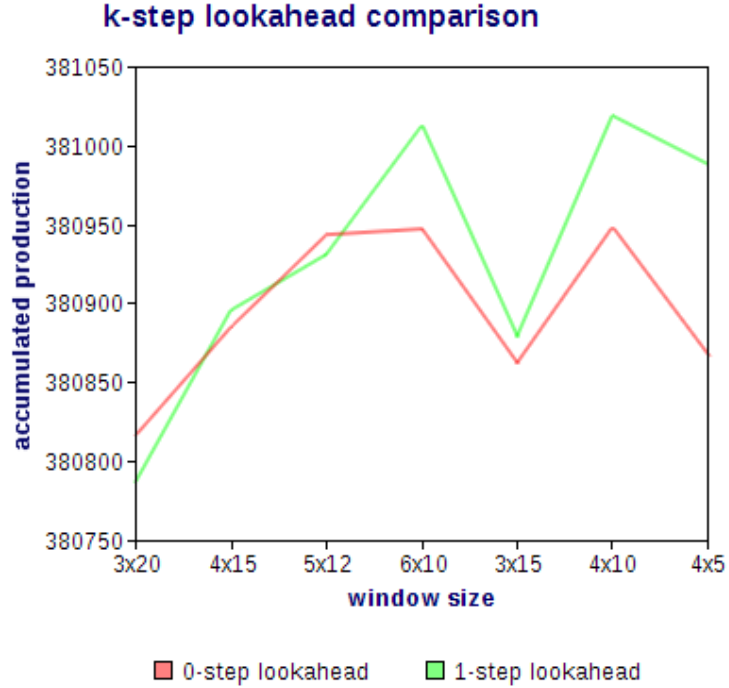


FIGURE 4.1: k-step lookahead comparison

Generally 1-step lookahead functionality boosts performance, as seen. Five out of seven different window sizes perform better with lookahead. Configurations with 15 elevation positions, 3×15 and 4×15 , present the smallest improvement whereas the smallest window setup, 4×5 , exhibit the biggest one. Moreover the two top performing window configurations, 6×10 and 4×10 , maintain leading results for both 0-step and 1-step lookahead options. In contrast there are two setups, 3×15 and 3×20 , that perform worse than the others for both 0-step and 1-step lookahead. Both of them have

3 azimuth positions which is a disadvantage in certain cases, as noted in Tables 4.3, 4.14.

Dynamic Programming methods comparison As mentioned in section 2.4 in page 13 another approach based on dynamic programming[33] was developed and tested under the same experimental configuration presented in this thesis. In the following table our results, in column WindowVI, and the ones of the aforementioned work are presented together.

TABLE 4.17: Results for different dynamic programming approaches. Values are in KWh and correspond to net energy gain.

Dataset		Single Axis				Dual Axis			
	Day	Typical	Myopic	SolarTrackingPI	WindowVI	Typical	Myopic	SolarTrackingPI	WindowVI
Historical	1	32.533	32.791	32.794	32.761	32.021	33.033	33.070	33.034
	2	52.036	52.042	52.046	52.017	51.624	52.326	52.360	52.321
	3	71.037	72.977	72.985	72.962	73.434	74.003	74.027	73.997
	4	11.623	11.738	11.754	11.730	11.465	11.788	11.822	11.782
Fictional	1	32.530	32.784	32.790	32.670	31.899	32.730	32.972	32.931
	2	52.034	52.040	52.045	51.961	51.515	52.034	52.247	52.166
	3	71.018	72.961	72.977	72.893	73.264	73.706	73.862	73.841
	4	11.615	11.729	11.751	11.706	11.411	11.567	11.747	11.726

Single axis case tracks the sun only in horizontal/azimuth direction while the other axis remains fixed throughout the day. The slope orientation is a set of different positions, one per day of study. These fixed orientations are calculated by searching the space of all available positions, taking into account the weather forecast for the next day. The columns correspond to the methods developed for the optimal sun tracking problem. Typical column refers to the commercially available sun tracking solutions that follow the sun based on an astronomical function as described in the beginning of this chapter. Also this column acts as a baseline for the rest of the results. The column WindowVI represent the results of our proposed algorithm. For each study case the best result from our experiments is presented. The WindowVI in single axis mode is a slight variation of our main algorithm. Slope positions are fixed throughout every experiment, hence $|\Lambda_{s_c}| = 1$ position in vertical direction. For azimuth dimension $|K_{s_c}| = 151$, meaning that all possible positions, as described in Table 3.2 in page 20, in horizontal direction are available in this case. It can be described as an extreme case of a single row window, $|K_{s_c} \times \Lambda_{s_c}| = |151 \times 1|$.

The remaining two columns are the dynamic programming methods presented in [33] and are described in Related work section 2.4 page 13. The datasets from the above table are:

Historical

- 1 → 20 March 2011

- 2 → 22 September 2012
- 3 → 21 June 2012
- 4 → 21 December 2008

Fictional

- 1 → 20 March 2011 with wind speed $60km/h$
- 2 → 22 September 2012 with wind speed $60km/h$
- 3 → 21 June 2012 with wind speed $60km/h$
- 4 → 21 December 2008 with wind speed $60km/h$

The above datasets are more extensively described in section 3.2.3 in page 33. For single axis results our method outperforms the typical sun tracker in all cases. In contrast, the other two dynamic programming methods perform better than ours. This can be explained due to the optimization process that is applied in fixed slope position selection for each method. Furthermore WindowVI was designed mainly for dual axis application. In this case the lack of freedom in vertical dimension acts as a disadvantage for our method.

In dual axis tracking our approach keeps performing better than the typical sun tracker. Compared against the remaining two methods, we see a marginal improvement over *myopic* in one out of four historical cases while a slight reduction is present in the rest three. In the case of fictional datasets, WindowVI stands between *myopic* and *SolarTrackingPI* in terms of performance. It is consistently better than *myopic* for each one of the four days. An explanation might rely on *myopic*'s lack of system's consumption consideration, a critical aspect for these four cases because of the enhanced wind speed presence. We could sum up our conclusions by stating that generally *SolarTrackingPI* present the top results, whereas our method, shares the next place with *myopic* in dual axis and follows it in single axis mode. Moreover, our method appears to be better than a typical sun tracker. It should be considered that the typical sun tracker(dual or single axis) technology is the dominant one in world market. Our results suggest that improvements can be made in the current trend.

Orientation examples

Apart from tabular results it is interesting to visualize the selected moves of the PVS through the day that produced these results. To this end we provide some graphical representations of our simulations. The best performing configuration is demonstrated for each case.

- **21 December [OVC]**

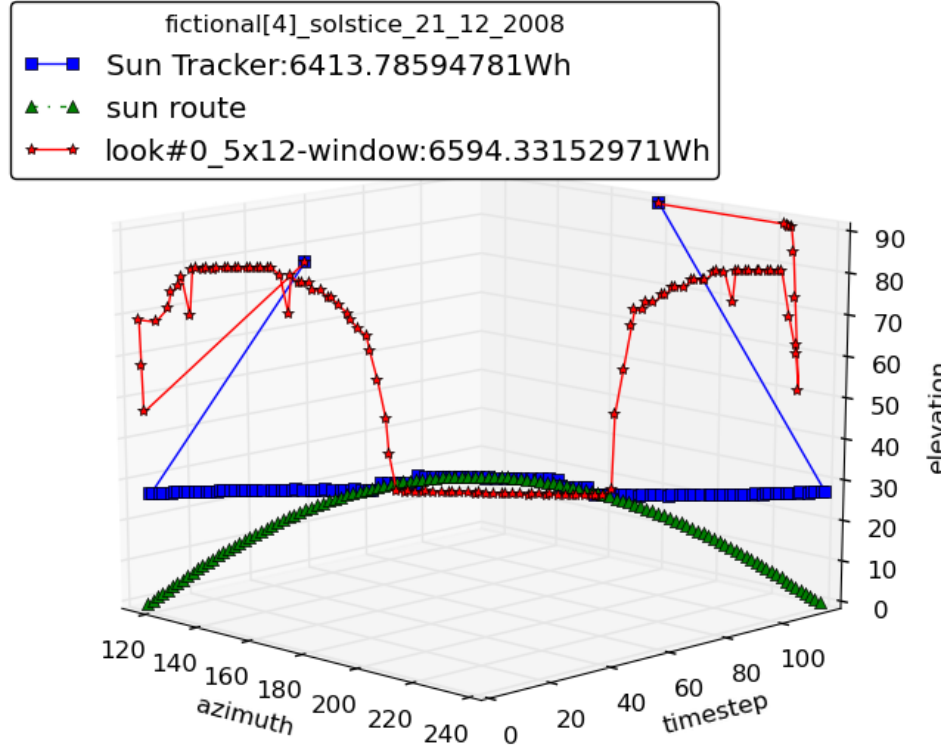


FIGURE 4.2: Graphical results for 21 December with complete overcast.

In Figure 4.2 we can see that the system's positioning is in parallel with the horizontal plane during most of the day. Only around noon it approaches sun's orientation.

- **21 June 2012**

The sky is completely clear during this summer day, hence our system's positioning strategy in Figure 4.3, which is similar to the typical dual axis tracker, comes as no surprise.

- **22 September 2012**

During this day scattered and mostly few clouds exist. Our system's orientation is similar to the typical tracker's one but in lower elevation degrees as seen in Figure 4.4.

-*Note:* The starting/ending point for all simulations is: azimuth= 180° (South), elevation= 90° (zenith)

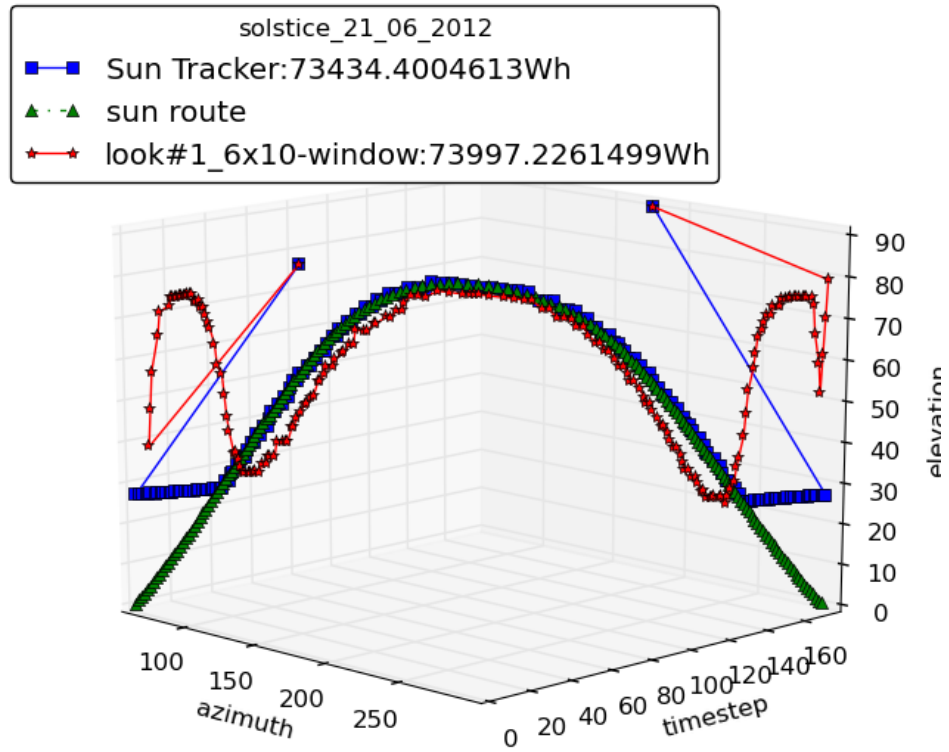


FIGURE 4.3: Graphical results for 21 June 2012 based on historical data.

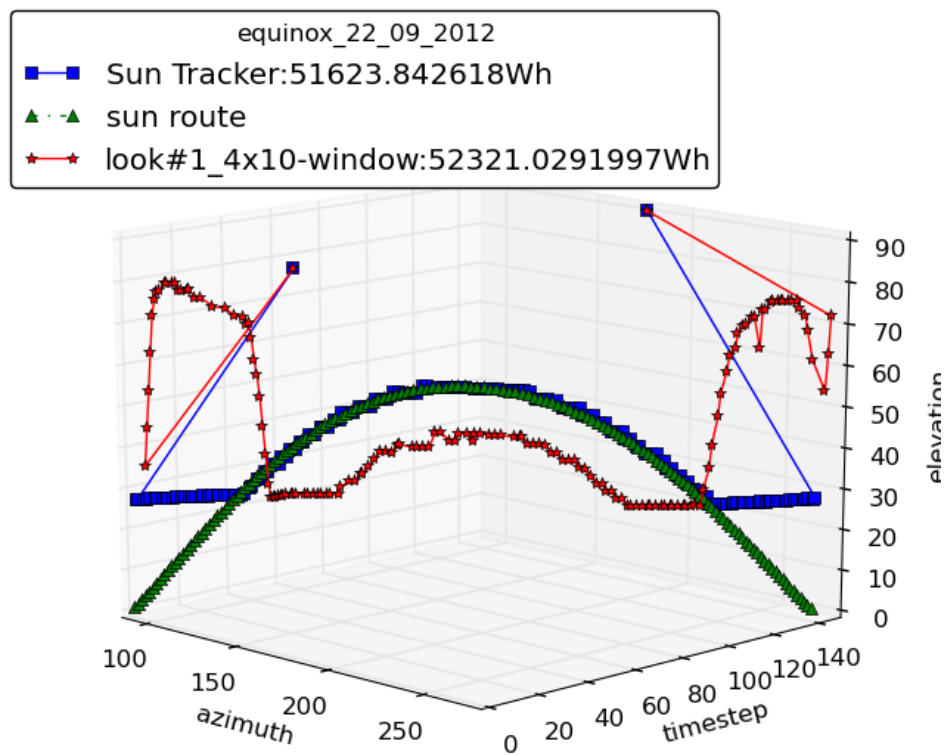


FIGURE 4.4: Graphical results for 22 September 2012 based on historical data.

Chapter 5

Renop- a Web User Interface

The approach presented in this thesis was also incorporated in a web based graphical tool, *Renop*. In order to visualize the results of the optimal sun tracking method that was proposed in this work, tabular results are exhibited against a typical dual axis sun tracker. Moreover the dynamic programming methods from [33], specifically *Next day optimal*, *Myopic* and *SolarTrackingPI*, were added as possible options for the user. All methods are available for testing over the datasets we created in section 3.2.3 in page 33. The page provides user input for WindowVI variables, such as k -step lookahead option and window $|K_{s_c} \times \Lambda_{s_c}|$ configurations as described in section 3.3 in page 35. For $k = [1, 2, 3]$ cases where computation time is rather large for user-interactive functionality, an email form is available for user input. When calculations are finished the results are sent automatically in the provided email address. A sample screenshot can be seen in figure 5.1.

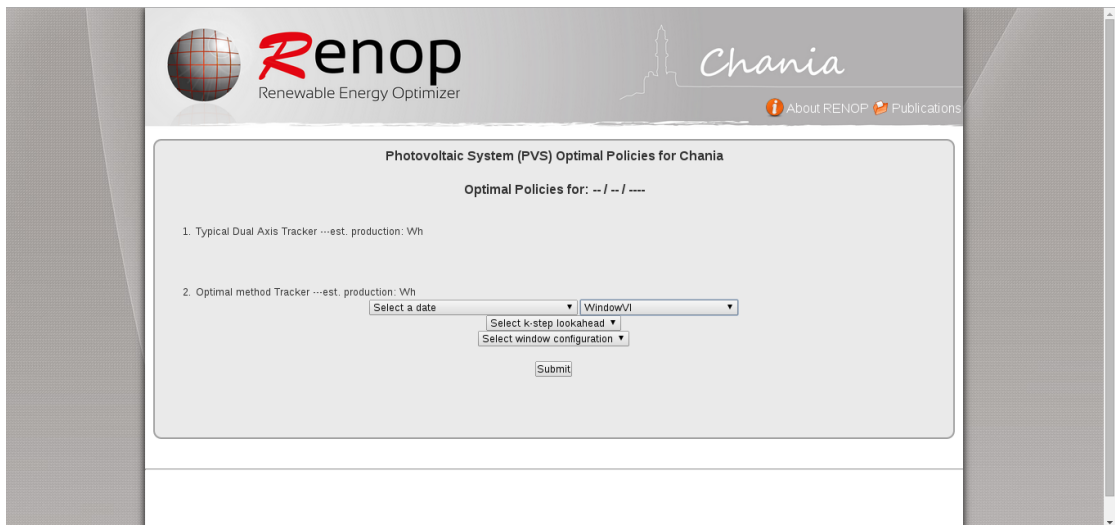


FIGURE 5.1: Renop - WindowVI option.

For the remaining three methods selection is pretty straightforward because none of them require user input as code variables. In the case of policy iteration algorithm, due to the intensive computation process, an email dialog box is provided with the same functionality as described above, in WindowVI case. A sample screenshot can be seen in figure 5.2.

FIGURE 5.2: Renop - policy iteration method.

After user input is submitted and appropriately parsed as arguments, our basic simulation code computes the system's optimal trajectory and net energy gain estimations. Subsequently a PHP¹ script handles the produced results and parses them in a Python script which updates the user interface accordingly. The final results of selected method are displayed against a typical sun tracker for comparison, as seen in figure 5.3. The

timesstep	time	azimuth	elevation
0	2008-12-21 05:30:00	118.8	27.0
1	2008-12-21 05:35:00	118.8	27.0
2	2008-12-21 05:40:00	120.6	27.0
3	2008-12-21 05:45:00	120.6	27.0
4	2008-12-21 05:50:00	122.4	27.0

timesstep	time	azimuth	elevation
0	2008-12-21 05:30:00	118.8	37.8
1	2008-12-21 05:35:00	118.8	41.4
2	2008-12-21 05:40:00	118.8	45.0
3	2008-12-21 05:45:00	118.8	41.4
4	2008-12-21 05:50:00	122.4	37.8

FIGURE 5.3: Renop results page.

information displayed is the estimated net energy production in *Wh* while the tables

¹<http://www.php.net/>

include the timestep of the specific orientation, corresponding datetime and azimuth elevation angles in degrees ($^{\circ}$). In detail,

- *timestep* are the discrete 5-minute timesteps as described in Table 3.2, page 20
- *time*, local time of Chania, `timezone('Greece/Athens')`
- *azimuth* angle in degrees ($^{\circ}$), with North: 0° and South: 180°
- *elevation* angle in degrees ($^{\circ}$), with zenith²: 90° .

Finally it should be mentioned that a *cache*-like functionality was created and added in site's algorithm. For every user given input the code is executed once; if the same configuration is requested later from another user a stored copy will be immediately available for viewing, thus speeding up the provided service.

²The zenith is an imaginary point directly "above" a particular location, on the imaginary celestial sphere. "Above" means in the vertical direction opposite to the apparent gravitational force at that location.

Chapter 6

Conclusions and future work

A new approach to solar tracking techniques was presented through this work. In order to improve the energy production simultaneously with system's consumption, the problem was modeled as an optimal control situation with sequential decision making. For this reason MDP framework was used and VI applied for approximating nearly optimal solutions corresponding to PVS orientation through the day. Similar approaches have never been developed before because the necessary reward model was non existant, but in this thesis we propose RENES estimator for this work. Furthermore a k -step look ahead functionality was added to the traditional VI scheme in order to improve results. The datasets evaluated cover a wide variety of weather patterns for different times of the year and the derived results show a clear improvement over the traditional solar tracking method. The configurations tested present different performance behaviour under different weather conditions as described in datasets. As presented in the previous Chapter 4 two of the tested configurations seem to outperform the rest ones.

As stated, the combination of solar tracking with optimal control strategies driven by AI methodologies has never been tried before so a wide area for future work has emerged. This could include different approaches for MDP solving with problem specific variations. Also an operational real world downscaled prototype could be build in order to correlate the simulated estimations with real measurements. Also computational cost reduction is of great significance. Algorithmic modifications and code optimization techniques, such as parallel computing, might be employed when applicable(e.g power production estimations from RENES backend). Finally, further improvements can be made in Renop web tool. In terms of user experience, server-side modifications could speed up execution times resulting faster response or better throughput if more algorithms are added in the future.

Bibliography

- [1] Stan Mark Kaplan, Fred Sissine, and Thecapitol Net. *Smart Grid: Modernizing electric power transmission and distribution; Energy independence, Storage and security; Energy independence and security act of 2007 (EISA); Improving electrical grid efficiency, communication, reliability, and resiliency; integrating new and renewable energy sources*. The Capitol Net Inc, 2009.
- [2] C.W. Potter, A. Archambault, and K. Westrick. Building a smarter smart grid through better renewable energy information. In *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, pages 1–5, March 2009. doi: 10.1109/PSCE.2009.4840110.
- [3] K. Moslehi and R. Kumar. A reliability perspective of the smart grid. *IEEE Transactions on Smart Grid*, 1(1):57–64, June 2010. ISSN 1949-3053. doi: 10.1109/TSG.2010.2046346.
- [4] A. Woyte, Vu Van Thong, R. Belmans, and J. Nijs. Voltage fluctuations on distribution level introduced by photovoltaic systems. *IEEE Transactions on Energy Conversion*, 21(1):202–209, March 2006. ISSN 0885-8969. doi: 10.1109/TEC.2005.845454.
- [5] G. M. Shafiullah, A.M.T. Oo, D. Jarvis, A.B.M.S. Ali, and P. Wolfs. Potential challenges: Integrating renewable energy with the smart grid. In *Universities Power Engineering Conference (AUPEC), 2010 20th Australasian*, pages 1–6, December 2010.
- [6] US DOE. *Grid 2030: A National Vision For Electricity's Second 100 Years*. 2003.
- [7] Sarvapali D. Ramchurn, Perukrishnen Vytelingum, Alex Rogers, and Nicholas R. Jennings. Putting the 'Smarts' into the smart grid: A grand challenge for artificial intelligence. *Commun. ACM*, 55(4):8697, April 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133825. URL <http://doi.acm.org/10.1145/2133806.2133825>.
- [8] Georgios Chalkiadakis, Valentin Robu, Ramachandra Kota, Alex Rogers, and Nicholas R. Jennings. Cooperatives of distributed energy resources for efficient

- virtual power plants. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '11, page 787794, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 0-9826571-6-1, 978-0-9826571-6-4. URL <http://dl.acm.org/citation.cfm?id=2031678.2031730>.
- [9] D. Pudjianto, C. Ramsay, and G. Strbac. Virtual power plant and system integration of distributed energy resources. *IET Renewable Power Generation*, 1(1):10–16, March 2007. ISSN 1752-1416. doi: 10.1049/iet-rpg:20060023.
- [10] Athanasios Aris Panagopoulos, Georgios Chalkiadakis, and Eftichios Koutroulis. Predicting the power output of distributed renewable energy resources within a broad geographical region. 2012.
- [11] Antonio Luque and Steven Hegedus. *Handbook of Photovoltaic Science and Engineering*. Wiley, July 2003. ISBN 9780471491965.
- [12] B.Y.H. Liu and R.C. Jordan. Daily insolation on surfaces tilted towards the equator. *ASHRAE Journal*, 3:5359., 1961.
- [13] Martin A. Green. *Solar cells: operating principles, technology, and system applications*. Prentice-Hall, 1982. ISBN 9780138222703.
- [14] P. J. Lunde. Solar thermal engineering: Space heating and hot water systems. *New York, John Wiley and Sons, Inc., 1980. 635 p., -1*, 1980. URL <http://adsabs.harvard.edu/abs/1980nyjw.book.....L>.
- [15] John A. Duffie and William A. Beckman. *Solar Engineering of Thermal Processes*. John Wiley & Sons, March 2013. ISBN 9781118418123.
- [16] Tian Pau Chang. The gain of single-axis tracked panel according to extraterrestrial radiation. *Applied Energy*, 86(78):1074–1079, July 2009. ISSN 0306-2619. doi: 10.1016/j.apenergy.2008.08.002. URL <http://www.sciencedirect.com/science/article/pii/S0306261908001839>.
- [17] Gladius Lewis. Optimum tilt of a solar collector. *Solar & Wind Technology*, 4(3):407–410, 1987. ISSN 0741-983X. doi: 10.1016/0741-983X(87)90073-7. URL <http://www.sciencedirect.com/science/article/pii/0741983X87900737>.
- [18] Hossein Mousazadeh, Alireza Keyhani, Arzhang Javadi, Hossein Mobli, Karen Abrinia, and Ahmad Sharifi. A review of principle and sun-tracking methods for maximizing solar systems output. *Renewable and Sustainable Energy Reviews*, 13(8):1800–1818, October 2009. ISSN 1364-0321. doi: 10.1016/j.

- rsr.2009.01.022. URL <http://www.sciencedirect.com/science/article/pii/S1364032109000318>.
- [19] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994. ISBN 0471619779.
- [20] Oguzhan Alagoz, Heather Hsu, Andrew J. Schaefer, and Mark S. Roberts. Markov decision processes: A tool for sequential decision making under uncertainty. *Medical Decision Making*, 30(4):474–483, July 2010. ISSN 0272-989X, 1552-681X. doi: 10.1177/0272989X09353194. URL <http://mdm.sagepub.com/content/30/4/474>. PMID: 20044582.
- [21] Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(12):181–211, August 1999. ISSN 0004-3702. doi: 10.1016/S0004-3702(99)00052-1. URL <http://www.sciencedirect.com/science/article/pii/S0004370299000521>.
- [22] Stephen P. Bradley, Arnoldo C. Hax, and Thomas L. Magnanti. *Applied mathematical programming*. Addison-Wesley Pub. Co., 1977. ISBN 9780201004649.
- [23] Richard Bellman. *Dynamic Programming*. Dover Publications, Mineola, N.Y, reprint edition edition, March 2003. ISBN 9780486428093.
- [24] R.A Howard. *Dynamic Programming and Markov Processes*. 1960.
- [25] F. R. Rubio, M. G. Ortega, F. Gordillo, and M. Lpez-Martnez. Application of new control strategy for sun tracking. *Energy Conversion and Management*, 48(7): 2174–2184, July 2007. ISSN 0196-8904. doi: 10.1016/j.enconman.2006.12.020. URL <http://www.sciencedirect.com/science/article/pii/S0196890407000064>.
- [26] M. Engin and D. Engin. Optimization mechatronic sun tracking system controller’s for improving performance. In *2013 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1108–1112, August 2013. doi: 10.1109/ICMA.2013.6618069.
- [27] A.H. ALQahtani, M.S. Abuhamdeh, Y.M. Alsmadi, and V.I. Utkin. Photovoltaic power optimization using sliding mode control with a two-axis tracking system. In *2013 IEEE Energytech*, pages 1–6, May 2013. doi: 10.1109/EnergyTech.2013.6645341.
- [28] I. Visa, D.V. Diaconescu, A. Duta, and V. Popa. PV tracking data needed in the optimal design of the azimuthal tracker #x2019;s control program. In *11th*

- International Conference on Optimization of Electrical and Electronic Equipment, 2008. OPTIM 2008*, pages 449–454, May 2008. doi: 10.1109/OPTIM.2008.4602447.
- [29] Xiaodong Zhang, Xiujuan Li, and Ke Lu. Research on an intelligent solar tracking system based on LPC2131. In *2012 3rd IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*, pages 429–432, September 2012. doi: 10.1109/ICNIDC.2012.6418789.
- [30] S. Seme, G. Stumberger, and J. Vorsic. Maximum efficiency trajectories of a two-axis sun tracking system determined considering tracking system consumption. *IEEE Transactions on Power Electronics*, 26(4):1280–1290, April 2011. ISSN 0885-8993. doi: 10.1109/TPEL.2011.2105506.
- [31] C. Alexandru and C. Pozna. Different tracking strategies for optimizing the energetic efficiency of a photovoltaic system. In *IEEE International Conference on Automation, Quality and Testing, Robotics, 2008. AQTR 2008*, volume 3, pages 434–439, May 2008. doi: 10.1109/AQTR.2008.4588958.
- [32] Giannis Kantaris. *Development of a sun-tracking system for maximizing the energy produced from photovoltaic arrays*. 2005.
- [33] Athanasios Aris Panagopoulos Georgios Chalkiadakis. Towards optimal solar tracking: a dynamic programming approach. Working Paper.
- [34] Deger tracker. URL http://www.degerenergie.de/tl_files/user-data/pdfs/products/trackers/2013_02_01_EN_Data%20Sheet_D60H_D80_D100_AS.pdf.
- [35] Sentinel solar. URL <http://stuartelectrical.ca/Sentinel%20%20Stewart%20Electrical%2010kw-tracker.pdf>.
- [36] Wattsun dual axis tracker. URL <http://arraytechinc.com/wp-content/uploads/NEW-DuraTrack-DA-Datasheet.pdf>.
- [37] Solar tracker d180, . URL <http://www.mechatron.eu/index.php/en/dual-axis-tracker>.
- [38] Solar tracker d170, . URL <http://www.mechatron.eu/index.php/en/dual-axis-tracker-d170>.
- [39] Titan tracker. URL http://www.titantracker.es/v_portal/inc/clicklink.asp?t=3&cod=2871&c=0&s=384253910.
- [40] Agoranet 2000 S. L. Dual-axis solar trackers for photovoltaic solar plants -design, manufacture and distribution- mecasolar. URL <http://www.mecasolar.com/pub/>

- [doc/File/ingl/2-axis-tracker-mecasolar-catalog.pdf](#). Three-phase, dual-axis solar trackers with 10-year guarantee on parts and workmanship. The trackers incorporate an anti-theft system for the photovoltaic solar panels.
- [41] M. Abouzeid. Use of a reluctance stepper motor for solar tracking based on a programmable logic array (PLA) controller. *Renewable Energy*, 23(34):551–560, July 2001. ISSN 0960-1481. doi: 10.1016/S0960-1481(00)00133-6. URL <http://www.sciencedirect.com/science/article/pii/S0960148100001336>.
- [42] P. Roth, A. Georgiev, and H. Boudinov. Cheap two axis sun following device. *Energy Conversion and Management*, 46(78):1179–1192, May 2005. ISSN 0196-8904. doi: 10.1016/j.enconman.2004.06.015. URL <http://www.sciencedirect.com/science/article/pii/S0196890404001554>.
- [43] R Akkaya and A. A Kulaksiz. A microcontroller-based stand-alone photovoltaic power system for residential appliances. *Applied Energy*, 78(4):419–431, August 2004. ISSN 0306-2619. doi: 10.1016/j.apenergy.2003.09.005. URL <http://www.sciencedirect.com/science/article/pii/S0306261903001934>.
- [44] Galileo tracker - doppler. URL <http://dopplergreenenergy.gr/%CE%BA%CE%B1%CF%84%CE%B7%CE%B3%CE%BF%CF%81%CE%AF%CE%B5%CF%82-%CF%86%CE%B2/galileo/>.
- [45] Henrik Svensmark and Eigil Friis-Christensen. Variation of cosmic ray flux and global cloud coveragea missing link in solar-climate relationships. *Journal of Atmospheric and Solar-Terrestrial Physics*, 59(11):1225–1232, July 1997. ISSN 1364-6826. doi: 10.1016/S1364-6826(97)00001-1. URL <http://www.sciencedirect.com/science/article/pii/S1364682697000011>.
- [46] Eigil Friis-Christensen and Henrik Svensmark. What do we really know about the sun-climate connection? *Advances in Space Research*, 20(45):913–921, 1997. ISSN 0273-1177. doi: 10.1016/S0273-1177(97)00499-7. URL <http://www.sciencedirect.com/science/article/pii/S0273117797004997>.
- [47] U.S. Office of the Federal Coordinator for Meteorological Services and Supporting Research (last). Surface weather observations and reports, 1998. URL <http://w3jjj.com/downloads/fmh1.pdf>.
- [48] J. A. Peterka and R. G. Derickson. Wind load design methods for ground-based heliostats and parabolic dish collectors. *NASA STI/Recon Technical Report N*, 93:15839, September 1992. URL <http://adsabs.harvard.edu/abs/1992STIN...9315839P>.

-
- [49] J. F. Orgill and K. G. T. Hollands. Correlation equation for hourly diffuse radiation on a horizontal surface. *Solar Energy*, 19(4):357–359, 1977. ISSN 0038-092X. doi: 10.1016/0038-092X(77)90006-8. URL <http://www.sciencedirect.com/science/article/pii/0038092X77900068>.
- [50] R Bruno. A correction procedure for separating direct and diffuse insolation on a horizontal surface. *Solar Energy*, 20(1):97100, 1978.
- [51] Benjamin Y. H. Liu and Richard C. Jordan. The interrelationship and characteristic distribution of direct, diffuse and total solar radiation. *Solar Energy*, 4(3):1–19, July 1960. ISSN 0038-092X. doi: 10.1016/0038-092X(60)90062-1. URL <http://www.sciencedirect.com/science/article/pii/0038092X60900621>.