Towards Optimal Solar Tracking: A Dynamic Programming Approach

Athanasios Aris Panagopoulos
Electronics and Computer Science,
University of Southampton, UK
ap1e13@ecs.soton.ac.uk

Georgios Chalkiadakis
Electronic and Computer Engineering,
Technical University of Crete, Greece
gehalk@intelligence.tuc.gr

Nicholas R. Jennings
Electronics and Computer Science,
University of Southampton, UK
nrj@ecs.soton.ac.uk

Abstract

The power output of photovoltaic systems (PVS) increases with the use of effective and efficient solar tracking techniques. However, current techniques suffer from several drawbacks in their tracking policy: *(i)* they usually do not consider the forecasted or prevailing weather conditions; even when they do, they *(ii)* rely on complex closed-loop controllers and sophisticated instruments; and *(iii)* typically, they do not take the energy consumption of the trackers into account. In this paper, we propose a policy iteration method (along with specialized variants), which is able to calculate near-optimal trajectories for effective and efficient day-ahead solar tracking, based on weather forecasts coming from online providers. To account for the energy needs of the tracking system, the technique employs a novel and generic consumption model. Our simulations show that the proposed methods can increase the power output of a PVS considerably, when compared to standard solar tracking techniques.

Introduction

In recent decades, a large number of photovoltaic systems (PVSs) have been integrated into the electricity grid. A PVS's power output depends mostly on the irradiance incident on the PV module and its operating temperature: in general, PVSs favor lower operating temperatures and greater levels of incident solar irradiance. In this context, *solar tracking (ST)* techniques can be used to orient the system towards the greatest possible levels of incoming solar irradiance. Depending on location and season, ST can increase the PVS power output by up to 100% (Mousazadeh et al. 2009).

The effectiveness of the ST technique used is therefore crucial for the overall efficiency of the PVS. Now, *active*[1] ST relies on electrical motors to move the PVS. The motors are driven by a controller that operates in a *closed-loop* or in an *open-loop* fashion (i.e., with or without making use of any feedback, respectively). Typical open-loop trackers are the *chronological* trackers, which follow the sun based on a chronological model of its motion (Reda and Andreas 2004). Although simple, they do not take into account the forecasted or prevailing weather conditions (e.g., the degree of cloud coverage). On the other hand, closed-loop controllers take the weather conditions into account, but they usually depend on expensive and sophisticated instruments (Lynch and Salameh 1990). On top of that, existing closed-loop controllers typically do not take into account the energy consumption caused by the tracking itself, nor do they consider the system's maintenance cost; they simply greedily turn the system towards the perceived highest irradiance values. For a review of ST techniques see (Mousazadeh et al. 2009).

In this work we develop novel *low-cost* ST techniques that can be used in both an *open-loop* or a *closed-loop* manner. We do not make use of expensive equipment or sensors, but the backbone of our approach is the estimation of the optimal trajectories a day before, based on weather forecasts coming from online providers for free. To this end, we employ a recently developed web tool, RENES (Panagopoulos et al. 2012), that predicts the power output of a PVS given available weather forecasts (www.intelligence.tuc.gr/renes). These predictions form the reward dynamics of a *policy iteration (PI)* technique we devise. The technique, *Solar Tracking Policy Iteration (STPI)*, alternatively optimizes over action sub-spaces. Although optimizing over sub-problems in an alternating fashion is a generally common concept elsewhere (Bezdek and Hathaway 2002), this is the first time that such an optimization technique is proposed for MDPs.

Importantly, the method makes use of a novel *tracking system consumption model* we develop (and which can be extended to account for maintenance and other costs). The method is appropriate for *dual-axis* (i.e., employing two axes of rotation) tracking, and is shown to be much more efficient than the, also sensor-less, *chronological* ST. We also provide four additional control methods: a PI method specialized for *single-axis tracking*, two near-optimal *myopic* methods (one specialized for *single-axis* and one for *dual-axis*) which we detail later, and a method that enables us to define the next-day-optimal positioning for any fixed-orientation (yet re-adjustable) PVS operating within the geographical region of a given weather station. The efficiency of the latter is higher than positioning the system according to *yearly-optimal* fixed-orientation estimates, and the method can be easily extended to define the *weekly-optimal* PVS orientation. Moreover, our methods are shown to improve the power output of a PVS even when compared to closed-loop *sensor-based* ST. Last but not least, all our methods come

[1] *Passive* ST, usually based on thermal expansion effects, is less common (Mousazadeh et al. 2009), and hence not considered here.

with guarantees of near-optimality.

Although dynamic programming can naturally find the optimal solution to the ST problem, this paper is the first to propose such an approach. This is probably due to the fact that an appropriate reward model had not been devised until now (due to the lack of free and ready-to-use power output estimates, and an appropriate consumption model). We resolve this issue, and thus contribute to the state-of-the-art, as follows: first, we employ the recent method of (Panagopoulos et al. 2012) to get PVS power output estimates;[2] and, second, we devise here, for the first time, a generic, parameterizable tracker power consumption model.

Our simulation results show that our approach outperforms all benchmark methods (i.e., chronological, sensor-based and/or fixed-orientation). Though we evaluate our approach via simulations involving the popular *azimuth-altitude dual axis trackers* (AADAT) and *vertical single axis trackers* (VSAT), we note that it can be used in conjunction with many other ST systems (e.g., *tilted axis*, or *horizontal single axis*). Finally, it is worth noting that our next-day policy comes complete with an expected PVS power output estimation. This is crucial for the smooth integration of PVSs into the electricity grid—since it is essential that short-term predicted PVS production estimates are available, notwithstanding their intermittent nature (Ramchurn et al. 2012).

Background

The total 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 (Luque and Hegedus 2011) :

$$G_T = G_B + G_D + G_R \tag{1}$$

Usually, the cosine effect is used to model the variations of the G_B component, as seen in Eq. 2 below:

$$G_B = G_B^{max} \cos\theta_s \tag{2}$$

where θ_s is the angle between the normal to the surface and the direction to the sun (as seen in Fig. 1) and G_B^{max} is the incident beam irradiance when the surface is oriented normally to the incoming radiation (i.e., $\theta_s = 0°$). G_B^{max} is the maximum beam irradiance that the PV module can orient to, and depends on weather conditions and solar position.

Now, the G_D component varies according to Eq. 3 which assumes that every point of the celestial sphere emits light with equal radiance (Liu and Jordan 1961):

$$G_D = G_D^{max}(1 + \cos\beta)/2 \tag{3}$$

where β is the inclination angle of the surface and G_D^{max} is the incident diffuse irradiance for $\beta = 0°$. G_D^{max} is the maximum diffuse irradiance that the PV module can orient to, and depends on weather conditions and solar position.

Finally, the G_R component is modeled by Eq. 4 which assumes that the ground is horizontal, of infinite extent, and reflects uniformly to all directions (Luque and Hegedus 2011):

$$G_R = G_R^{max}(1 - \cos\beta) \tag{4}$$

[2]We note here, that one could alternatively use the work of (Chakraborty et al. 2012), which, however, unlike RENES, does not come with a ready-to-use web tool, and requires the availability of historical PVS-specific production output data.

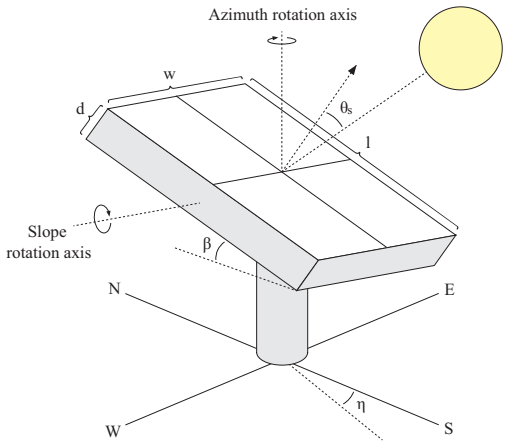

Figure 1: Abstract AADAT (in VSAT β is fixed)

where G_R^{max} is the maximum reflected irradiance that the PV module can orient to (for $\beta = 90°$), and depends on weather conditions and solar position.

In our work we focus on both *AADAT* and *VSAT* tracking systems. The AADAT has two degrees of freedom, rotating over a slope (elevation) and an azimuth axis. An abstract AADAT is illustrated in Figure 1. The VSAT rotates only over the azimuthal axis, while its slope angle is kept fixed.

Typically, the movement allowed in tracking systems is constrained within a certain range in both the azimuthal and elevation axis. We henceforth denote the allowed azimuthal and elevation angular range by r_{Az} and r_{Sl} respectively.

The possible slope and azimuth orientations of a dual axis system consists of a discrete number of possible positions within the allowed range at each rotation axis, depending on the tracker step size. Now, a misalignment of $\pm 1°$ causes only a minor drop of $\sim 0.015\%$ in the incident beam irradiance G_B (cf. Eq. 2). Thus, small misalignments are not a concern for typical commercial systems.

The controller step-size (i.e., the system's minimum angular displacement) θ gives rise to two sets of distinct possible orientation positions for the PVS (one such set per axis of movement). We denote these by K, the set of azimuth orientation positions, and by Λ, the set of possible positions on the elevation axis. In particular, we have $|K| = \lfloor r_{Az}/\theta \rfloor + 1$ and $|\Lambda| = \lfloor r_{Sl}/\theta \rfloor + 1$. The time required for a minimum displacement θ to occur is denoted by δ; its value is assumed constant in our model, in order to maintain a fixed mean angular velocity for every minimum displacement.

Now, the controller requires some time to interact with the PVS, and it takes the system some time to execute the controller commands. For simplicity, the controller in our model is synchronous, meaning that any two consecutive controller-system interactions are separated by a fixed-length time interval Δ. A natural choice is to pick a Δ length that is sufficient to move the PV panel at any orientation starting from an arbitrary position (i.e., $\Delta \geq \delta \cdot max(|K| - 1, |\Lambda| - 1)$), and small enough so as the environmental conditions do not change abruptly during this interval.

Hence, the operation time of a PVS is naturally divided

into a number of equal time intervals of length Δ each. Letting Δ_{Day} stand for the day-length (i.e., the time between sunrise and sunset), the required controller-system interactions are contained in a set I, with $|I| = \lceil \Delta_{Day}/\Delta \rceil$. Note that each interaction corresponds to a specific *time-step* or *interaction id* τ; and that these interactions are sufficient to orient the PV panel to any orientation right after the sunrise, and to move it back to the initial position after the sunset.

A Dynamic Programming Approach

The aim of our work is to calculate optimal ST trajectories for the day ahead, based on available weather forecasts—which can actually come from online providers for free. To this end, we employ dynamic programming and, in particular, an intuitive *policy iteration* technique (and variants).

Defining the MDP

The problem is naturally modeled as a fully observable, finite-horizon, discrete-time Markov decision process (MDP) corresponding to an $\langle S, A, P, R \rangle$ tuple as follows:

First, S is a finite set of *states*, where each state $s \in S$ corresponds to a tuple $\langle \kappa_s, \lambda_s, \mathbf{w}_s \rangle$ with $\kappa_s \in [1, |K|]$ denoting the azimuth orientation position, $\lambda_s \in [1, |\Lambda|]$ denoting the slope orientation position, and $\mathbf{w_s}$ standing for the vector of *stochastic weather condition variables* which are required to calculate the MDP reward dynamics (i.e., prevailing wind speed and direction, relative humidity, temperature, and sky conditions). As such, $|S| \geq |K \times \Lambda|$. The value of each state depends on the $\tau \in [1, |I|]$ time-stamp at which the state is visited—that is, $|I|$ represents the horizon for our problem. Note that the time-stamp τ at which s is visited is sufficient to extract all necessary information regarding the *non-stochastic environmental conditions* (i.e., the solar position angles) relevant to s and τ. Therefore, these do not have to be explicitly included in the state representation.

Then, A is a finite set of *actions*, with each action $a \in A$ positioning the PVS to some specific orientation. Thus, a corresponds to tuple $\langle \kappa_a, \lambda_a \rangle$, with $\kappa_a \in [1, |K|]$ and $\lambda_a \in [1, |\Lambda|]$. Hence, we have $|A| = |K \times \Lambda|$.

The *transition model* P defines the $P(s, a, s')$ probability that taking action $a = \langle \kappa_a, \lambda_a \rangle$ in state s will lead to s'. Thus, given a particular action a at a state s, for the successive state s' we will have: $\kappa_{s'} = \kappa_a$, $\lambda_{s'} = \lambda_a$; while the transition probabilities will depend entirely on the $P(\mathbf{w'_s}|\mathbf{w_s})$ ones. Note that in the case that $\mathbf{w'_s}$ is independent of $\mathbf{w_s}$ we will not have a *probabilistic transition model*, but rather a *probabilistic reward model*. Hence, in that case, $\mathbf{w_s}$ can be omitted from the state representation and expected reward values can be extracted directly from the controller interaction id, τ, at which s is visited. The same holds if only non-probabilistic weather forecasts are available.

Finally, R is a *reward model* determining the $R_a(s, s')$ reward received for a transition from state s to s' after taking action a. This reward is the energy produced during the time between two consecutive controller interactions, minus the energy consumed due to the movement of the tracker throughout this interval. Thus:

$$R_a(s, s') = Prod(s, s') - Cons(s, s') \tag{5}$$

where $Prod(s, s')$ and $Cons(s, s')$ are functions estimating the energy produced and consumed as a result of the PVS system moving from s to s' (after some action a taken at s).

Calculating $Prod(s, s')$ is straightforward, assuming that the PVS power output is steady throughout a time interval Δ between two consecutive controller interactions:

$$Prod(s, s') = \frac{(Pwr(s) + Pwr(s'))}{2}\Delta \tag{6}$$

where $Pwr(s)$ stands for the PVS power output at state s. In our work, the $Pwr(s)$ estimates are provided by RENES, given the PVS orientation (i.e., κ_s,λ_s), the particular time of day (derived based on τ and used to estimate the solar position angles), the (fixed for a given system)) PV characteristics, as well as the stochastic weather conditions in $\mathbf{w_s}$.

Consumption model A distinct contribution of our work is the construction of a generic and parameterizable solar tracker consumption model. With an arbitrary displacement corresponding to an aggregation of minimum angular displacements on each one of the rotation axes, we calculate the consumption of an arbitrary displacement as the (efficiency-weighted) sum of the consumptions corresponding to these minimum angular displacements. Now, in order to maintain a fixed mean angular velocity for every minimum angular displacement θ, every such θ is assumed to follow a simple trapezoid motion profile with three motion phases (of equal time duration): *(1)* an *angular acceleration phase*, *(2)* a *constant angular velocity phase*, and *(3)* an *angular deceleration phase*. As such, the consumption for θ is calculated as the sum of the consumption for all three motion types in sequence. Then, the system consumption $Cons(s, s')$ is:

$$Cons(s, s') = \frac{1}{c_{eff}}\left(\sum_{1}^{|\kappa_s - \kappa_{s'}|} Cons_{\theta}^{az} + \sum_{1}^{|\lambda_s - \lambda_{s'}|} Cons_{\theta}^{sl} \right) \tag{7}$$

where c_{eff} stands for the efficiency factor of the tracking system. This corresponds to the mean efficiency of the motors, multiplied by the mean efficiency of the gears, and is further reduced to best fit all other secondary losses of the system during a displacement. $Cons_{\theta}^{az}$ and $Cons_{\theta}^{sl}$ represent the consumption of every minimum angular displacement θ over the azimuth and slope (elevation) axis respectively. Their values are calculated by the following equations:

$$Cons_{\theta}^{az} = \sum_{\mu=1}^{3} (\alpha_\mu \mathcal{I}_{A_{(\theta,\mu)}} - T^{w}_{A_{(\theta,\mu)}})\theta_\mu \tag{8}$$

$$Cons_{\theta}^{sl} = \sum_{\mu=1}^{3} (\alpha_\mu \mathcal{I}_{S} - T^{w}_{S_{(\theta,\mu)}})\theta_\mu \tag{9}$$

where θ_μ and α_μ stand for the angular displacement and acceleration for each one of the motion phases, and can be computed as $\theta_1 = \theta_3 = \theta_2/2 = \theta/4$, and $\alpha_1 = -\alpha_3 = 9\theta/2\delta^2$ and $\alpha_2 = 0$. Now, $\mathcal{I}_{A_{(\theta,\mu)}}$, $\mathcal{I}_S$, $T^{w}_{A_{(\theta,\mu)}}$ and $T^{w}_{S_{(\theta,\mu)}}$ stand for the *moment of inertia* and wind *torque*, for the azimuth and slope axis respectively.

For the slope rotation, the moment of inertia is independent of the azimuthal orientation, and, assuming the panel is a cuboid, can be given by $\mathcal{I}_S = \frac{m}{12}(l^2 + d^2)$ (Myers 1962), where m stands for the mass of the panel, and l and d for the length and thickness of the panel as seen in Fig. 1. Note, however, that the azimuthal motion occurs simultaneously with the slope one. Hence, T_S^w, T_A^w and $\mathcal{I}_A$ are not constant during the motion. Nevertheless, due to the very small displacement corresponding to each motion phase, these quantities were assumed constant and equal to their value at the beginning of each motion phase.

For the azimuthal rotation, the moment of inertia depends on the slope orientation. Assuming that the panel is a cuboid, the moment of inertia for the azimuthal rotation given a particular slope angle β can be computed as follows:

$$\mathcal{I}_A = \frac{m}{12}\left(l^2\cos^2(\beta) + d^2\sin^2(\beta) + w^2\right) \qquad (10)$$

where w stands for the *width* of the panel.

We also modeled the PVS aerodynamics, estimating the torque on the rotation axes due to the incident wind as $T_X^w = \frac{1}{2}\rho w l^2 V^2 c_X$, where $X \in \{A, S\}$, ρ denotes the *air density*, V the prevailing *wind speed*, and c_A and c_S denote the non-dimensionalized slope and azimuth *moment coefficients*, respectively. These coefficients depend on the orientation of the system. The air density was estimated based on the *local pressure*, the *relative humidity*, and the *temperature*, based on standard meteorological equations (Picard et al. 2008). The moment coefficients are calculated based on *wind direction* and *system orientation*, as in (Roos 2012).

Optimal Solar Tracking

With the above MDP at hand, the *optimal ST policy* can be derived by solving the corresponding Bellman optimality equation (Sutton and Barto 1998). However, due to the size of the state and action spaces (typically $|I|\cdot|S|\cdot|A| > 4$Bn, without even considering $\mathbf{w_s}$),[3] the optimal tracking policy for the day-ahead cannot be computed exactly in realistic settings applications. Rather, it can only be approximated. To this end, we have devised several approximation methods, which we now proceed to describe.

Approximation Methods

We now describe the approximation techniques we developed in order to compute effective ST policies.

Solar Tracking Policy Iteration method (STPI)

We devised a policy iteration (PI) approximation technique to compute a beneficial ST policy. The technique interweaves two distinct PI procedures, which are used in an alternating fashion. The first PI procedure, *SlopePI*, considers an arbitrary input policy for the above MDP, e.g., a myopic one. It then attempts to improve that policy, in a usual PI fashion, but assuming a *fixed azimuthal policy*, π_κ. Given this fixed π_κ policy, it computes the respective optimal slope-positioning policy, π_λ. The output policy is then fed in a second PI algorithm, which estimates an optimal (given π_λ) azimuth-positioning policy, π_κ. The process repeats until convergence, or until some computational or time limit is reached. By combining the derived policies computed for each axis, we can derive a ST policy. The same PI algorithm can be readily employed for single axis tracking, with the action selection process for the static axis (the slope one, in the case of VSAT) considering only a set of fixed possible orientations for the whole motion (so as to estimate the best possible fixed slope angle for VSAT tracking during the next day). The overall PI technique is shown in Alg. 1, while Alg. 2 describes the PI process to derive a slope policy (the PI for deriving an azimuthal policy is entirely similar). Note that STPI effectively alternates between solving MDPs with state-action spaces which are orders of magnitude smaller than that required by the original problem formulation. Though there are no formal guarantees for convergence to the optimal policy,[4] the technique is intuitive, and exhibits good behavior in practice. We note here again that, surprisingly enough, this is the first time that an approach that alternatively optimizes over MDP action subspaces is proposed for optimal policy approximation.

That said, the choice of the initial policy used is crucial for the efficiency of any policy iteration algorithm (Sutton and Barto 1998). The initial policy we use is a *myopic* one, which maximizes power output alone (disregarding any associated repositioning costs). Now, the tracking consumption of a PVS is a very small fraction of its production (typ-

Algorithm 1 "Alternating" Policy Iteration for ST

procedure STPI(π)
  Initialize π_λ and π_κ based on π
  while π_λ and π_κ are not stable **do**
    $\pi_\lambda \leftarrow$ SLOPEPI(π_λ, π_κ)
    $\pi_\kappa \leftarrow$ AZIMUTHPI(π_κ, π_λ)
  Derive π' by combining π_κ and π_λ
  return π'

Algorithm 2 Slope Policy Iteration

procedure SLOPEPI(π_λ, π_κ)
  while π_λ is not stable **do**
    for all $\tau \in I$ in descending order **do**
      for all $s \in S$ that can emerge based on π_κ at τ **do**
        $a \leftarrow \langle \kappa_a = \pi_\kappa(s,\tau), \lambda_a = \pi_\lambda(s,\tau)\rangle$
        $V_\tau(s) \leftarrow \sum_{s'} P(s,a,s')\left(R_a(s,s') + V_{\tau+1}(s')\right)$
    for all $\tau \in I$ (in any order) **do**
      for all $s \in S$ that can emerge based on π_κ at τ **do**
        $\pi_\lambda(s,\tau) \leftarrow \operatorname{argmax}_\lambda \sum_{s'} P(s,a,s')(R_a(s,s')+$
        $V_{\tau+1}(s'))$, where $a = \langle \kappa_a = \pi_\kappa(s,\tau), \lambda_a = \lambda\rangle$
  return π_λ

[3] In more detail, for a day with 12 daylight hours (i.e., $\Delta_{Day} = 12\ hours$); a typical system (like the one considered in our evaluation) with $r_{Az} = 270^\circ$, $r_{Sl} = 63^\circ$ and $\theta = 1.8^\circ$ (at each axis); and a control interval of 5 min (i.e., $\Delta = 5\ min$), there will be: $|I| = \lceil \Delta_{Day}/\Delta \rceil = 144$, $|A| = |K|\cdot|\Lambda| = \lfloor r_{Az}/\theta \rfloor + 1 \cdot \lfloor r_{Sl}/\theta \rfloor + 1 = 5,436$, $|S| \geq |K|\cdot|\Lambda| \Rightarrow |S| \geq 5,436$, and, hence, $|I|\cdot|S|\cdot|A| \geq 4,255,213,824$.

[4] Intuitively, however, since each iteration improves on the current policy, STPI is expected to converge to a fixed point.

ically less than 1% (Mousazadeh et al. 2009)). As such, the *myopic policy* is essentially a *near-optimal* one, achieving over 99% of the optimal performance. Moreover, any gains achieved by a method that lowers consumption is typically accompanied by costs due to lower production. Hence, the near-optimality guarantees of the myopic policy are even stronger; and they are *inherited by STPI*, as any derived improved policy cannot be worse than the initial one. We now describe how to derive the myopic policy.

Myopic method

We define the *myopic policy* in this domain as a method that maximizes power generation only. As such, we can modify the MDP reward function to account for the PVS power output only: $R_a(s, s') = Prod(s, s')$. Given the fact that all possible PVS orientations are accessible from any state, it is clear that the optimal policy in this modified MDP is equivalent to the one that chooses the action that gives the maximum expected reward for the next time interval.

Now, the power output of a PVS increases proportionally to incident irradiance. In order to maximize the incident G_T we need to maximize the sum $G_B + G_D + G_R$ as seen in Eq. 1. Moreover, as seen in Eqs. 3 and 4, G_D and G_R components vary from their maximum based only on the slope angle of the PV module. On the other hand, Eq. 2 illustrates that 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. In particular, G_B reaches its maximum as the incident angle reaches zero (i.e., when the azimuth and slope angle of the PV module are the same as the azimuth and slope angle of the sun). As such, fixing the azimuth angle to follow the sun azimuth, ensures that we are always able to track the maximum G_T (for any weather conditions). The only thing that we need to do is to optimize the PVS slope angle at every time step, so that we balance the G_T components and get the maximum expected G_T. For (vertical) 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).

Next-Day-Optimal Fixed-Orientation

In the context of this work, we also propose and calculate the *next-day-optimal fixed PVS orientation*, by simply evaluating the whole space of possible orientations, given the next-day weather prediction. The derived orientation is suitable for any fixed-orientation (yet re-adjustable) PVS operating within the geographical region of a given weather station. Moreover, next-day-optimal fixed positioning can also be used by trackers in the case of scheduled power cuts. We note that this method can also be extended for weekly-optimal or some hours-optimal positioning, as needed.

Simulation Experiments

In order to evaluate our methods we simulated a PVS located at Chania, Crete, and estimated its output energy gain from employing each one of our methods. We chose Crete for our evaluation due to the great degree of PVS penetration on this sunny Greek island.[5] Moreover, this choice ensures that RENES provides accurate PVS power output predictions (Panagopoulos et al. 2012). That said, we note that considering locations with lower sunshine and greater wind speeds for our evaluation, would only favor our methods (as suggested by our evaluation results with "fictional" weather data below). Now, our simulations were performed for 8 different daily weather patterns; 4 of them corresponding to actual, real days, and 4 of them fictional. Moreover, we compared our methods against additional baseline methods we implemented for this purpose: *chronological VSAT* and *AADAT*, and a *yearly optimal fixed-orientation* system.

In more detail, the *chronological* AADAT calculates the sun positions as prescribed in the work of (Reda and Andreas 2004), and then orients the PVS so as to point towards the sun, irrespective of weather conditions. For the *chronological* VSAT, we used the same procedure to calculate the PVS azimuthal positions; while the slope angle was fixed to its yearly optimal value for VSAT tracking, given the location's latitude, as prescribed in (Li et al. 2011). Finally, we used the equations provided in (Chang 2009) to calculate the yearly optimal slope position for *fixed-orientation* south-facing panels at our location of interest.

The Simulated Photovoltaic System and Dataset

We modeled a typical $72m^2$ system (i.e., $w = 6.0m$, $l = 12.0m$, $d = 0.20m$) with 270° of azimuthal motion range, and 63° of elevation motion range. The system weight was set to $2500kg$. The modeled system was limited to provide a *step-size* of $\theta = 1.8^\circ$ at each axis, which could lead to a maximum misalignment of $\arccos(\cos^2(\theta))/2 \simeq 1.27^\circ$, corresponding to a G_B drop of $\sim$0.025%. The time δ required for a minimum displacement θ to occur, was set to $1s$, and the interval between two consecutive controller interactions was set to $\Delta = 5min$. As the efficiency of the motors and gears depends on the speed and load at all times (Burt et al. 2008), we used a mean efficiency of 30% for both[6] which leads to consumption that is close to the reported practical value for such systems (Mousazadeh et al. 2009).

For the purposes of our research, archival weather data was accumulated from the *weather underground* website (www.wunderground.com) regarding four different days at our location of interest. Specifically, we accumulated archival weather data for the $20/03/2011$ equinox, the $22/09/2012$ equinox, the solstice of $21/06/2012$, and the solstice of $21/12/2008$. These days are noted from now on as day 1, 2, 3 and 4 respectively. As there is a 30-minute gap between consecutive archival weather data, we use linear interpolation to meet the model's time interval requirements. Furthermore, as we are interested in the prevailing conditions within a $\Delta = 5min$ interval, all variables are assumed constant and equal to their mean value within that interval.

In particular, the acquired meteorological variables are:

[5] $\sim 60MW$ of installed PV power, corresponding to $\sim$7% of Crete's energy production and $\sim$70% of the total Greek islands' installed PV power (Public Power Corporation of Greece).

[6] Based on data provided at www.acosolar.com; and users.ece.utexas.edu/~valvano/Datasheets.

Dataset	Day	Fixed-Orientation		Single Axis ST (VSAT)			Dual Axis ST (AADAT)		
		Year-Opt	Next-Day-Opt	Chronological	Myopic	STPI	Chronological	Myopic	STPI
Real	1	31.520 (-)	**32.448** (-)	32.533 (.027)	32.791 (.019)	**32.794** (.015)	32.021 (.061)	33.033 (.078)	**33.070** (.037)
	2	49.736 (-)	**50.275** (-)	52.036 (.029)	52.042 (.028)	**52.046** (.023)	51.624 (.063)	52.326 (.087)	**52.360** (.049)
	3	67.301 (-)	**68.921** (-)	71.037 (.039)	72.977 (.057)	**72.985** (.048)	73.434 (.091)	74.003 (.106)	**74.027** (.080)
	4	11.736 (-)	**11.748** (-)	11.623 (.019)	11.738 (.031)	**11.754** (.010)	11.465 (.037)	11.788 (.059)	**11.822** (.021)
Fictional	1	31.520 (-)	**32.448** (-)	32.530 (.030)	32.784 (.026)	**32.790** (.019)	31.899 (.183)	32.730 (.381)	**32.972** (.121)
	2	49.736 (-)	**50.275** (-)	52.034 (.031)	52.040 (.030)	**52.045** (.023)	51.515 (.172)	52.034 (.379)	**52.247** (.156)
	3	67.301 (-)	**68.921** (-)	71.018 (.059)	72.961 (.074)	**72.977** (.055)	73.264 (.261)	73.706 (.404)	**73.862** (.237)
	4	11.736 (-)	**11.748** (-)	11.615 (.026)	11.729 (.041)	**11.751** (.010)	11.411 (.090)	11.567 (.280)	**11.747** (.032)

Table 1: Simulation results (all values are in kWh, and correspond to PVS net energy gain; tracking consumption in parenthesis).

relative humidity, *temperature*, *wind speed*, *wind direction*, and *qualitative cloud coverage observations* (appropriately transformed to quantitative values, as in Panagopoulos et al. (2012)). The observed general weather patterns of the days examined is as follows. The general pattern for *Day 1* consists of several transitions in the cloud coverage levels (moving from mostly sunny to scattered clouds, then to mostly cloudy, and back to mostly sunny); the general weather pattern for *Day 2* consists of a simple transition in the cloud coverage levels, from mostly cloudy to clear sky; there was a clear sky throughout *Day 3*; while *Day 4* was a day with (almost) full cloud coverage.

Now, though only four, these days exhibit weather patterns that are quite distinct from each other, thus enabling us to test the model under a wide range of meteorological conditions at our location of interest. However, throughout these four days, the prevailing wind speed value was quite low. Given the fact that power consumption grows with *high* prevailing wind speed, we decided to test the system with fictional wind data. We thus created four additional, "fictional" days, with exactly the same weather conditions as their real counterparts—apart from the wind, whose speed was set to *60 km/h*. That value corresponds to a typical maximum wind speed that a PVS can withstand without a need to orient itself to a safe position (Peterka and Derickson 1992).

In the absence of probabilistic weather forecasting reports (and respective online providers) we used deterministic archival weather data for both the weather predictions and ground-truth. As such, in our experiments, the accuracy of the weather forecasts does not affect the efficiency of our methods. We note here, that in the absence of weather prediction uncertainty, the evaluation results of the proposed low-cost *Myopic* method are equivalent to a tracking system where an expensive sensors arrangement along with a closed-loop controller is used to orient the solar panel towards the maximum incident solar irradiance. As such, by comparing *STPI* against *Myopic* we effectively compare our sensor-less, low-cost STPI against sensor-based ST.

The Results

The evaluation results of the experiments are collectively reported in Table 1. All energy values are in kWh, and correspond to PVS net energy gain. Tracking consumption is also reported inside parenthesis (when applicable).

In all experiments, our methods clearly outperform the baseline ones. It is also worth noting that, in general, as the system's degrees of freedom are increased, so do the positive system efficiency effects from using our methods (i.e., compared to fixed-orientation systems, the net energy gain increases when using *Myopic* or *STPI* with one rotation axis; and it increases even more when using these methods with two axes of rotation). By contrast, the benefits from using chronological ST often decrease when moving from fixed-orientation to one and, further on, to two rotation axes, as the additional system abilities are not fully exploited.

Regarding the methods' individual performance, not surprisingly, *next-day optimal fixed-orientation* significantly outperforms the yearly optimal one. In addition, *Myopic* gives a significant advantage over chronological tracking, in both VSAT and AADAT, as it also considers the weather conditions. At the same time, *STPI* does consistently better than *Myopic*, even though not by a wide margin. This low improvement margin is not surprising: in an appropriately designed, sizable PVS, like the one used in our simulations, the tracking consumption is *much* lower than the energy produced (less than 1% (Mousazadeh et al. 2009)). Thus, the net energy gains achieved by methods that take consumption into account, are not expected to differ dramatically from those achieved by methods that maximize power generation notwithstanding consumption needs. This fact is confirmed from our evaluation results: an improvement from using *STPI* instead of *Myopic* is present in all days and tracking systems, but is more substantial for high prevailing wind speeds, and especially for dual-axis tracking. However, over time (i.e., within a long operating time window and/or for clusters composed of many PVSs put together), even small improvements like the ones observed are significant. Even for an average-sized PV park of 2MW nominal power, one would be able to, annually, gain over €1500 more by using *STPI*, compared to *Myopic* (and over €10000 compared to chronological AADAT). Moreover, smaller PVSs (or not very efficiently designed ones) would exhibit a higher consumption over production ratio, and thus a higher expected improvement from using *STPI*, which attempts to approximate the optimal policy. Of course, the net gain is expected to further improve if the actual optimal policy is computed. However, *Myopic* is already near-optimal, as argued above; and, when compared to *Myopic*, *STPI* is shown in our simulations to already be achieving higher net energy gains and substantially lower consumption (of up to $\sim$90% reduction).

As a final note, *STPI* is expected to yield increased benefits when one considers a more detailed consumption model. In particular, the tracking cost is not limited to the motor consumption; there is also the maintenance cost, which should, ideally, also be taken into account. Moreover, real-world buy and sell energy prices will most probably have different values. These requirements can be readily incorporated in our model, by simply modifying Eq. 5. Specifically, for a grid-connected PVS, Eq. 5 can be replaced by:

$$R_a(s, s') = Prod(s, s')p^s - Cons(s, s')p^b - (t^{op}c_m) \quad (11)$$

Here, c_m denotes maintenance cost given operating time t^{op}, and p^s and p^b denote the sell and buy energy prices.

Conclusions and Future Work

In this work, we formulated ST as a dynamic programming problem. The exact solution to this problem would provide the optimal ST trajectories. However, for reasons of computability, we approximate the optimal solution by a policy iteration method (along with specialized variants) which utilizes available weather forecasts—that can actually come from online providers for free. Importantly, our methods make use of a generic and parameterizable tracker power consumption model we put forth. All our methods come with near-optimality guarantees, and we demonstrated their efficiency against commonly employed conventional ST techniques (i.e., chronological, sensor-based, and/or fixed-orientation). As such, our methods can serve as the basis for the development of web-based tools for efficient predictive ST. Future work includes incorporating additional details in our model (as, e.g., demonstrated by the suggested use of Eq. 11 above). Furthermore, we aim to modify our methods to account for additional tracking systems (such as pole trackers), and to consider weather forecasts updates throughout the operation day, in real-time.

Acknowledgements We thank Andonis Mitidis (Univeristy of Florida) and Kostas Lagogiannis (University of Edinburgh) for comments and fruitful discussions. We also acknowledge funding from the UK Research Council for the ORCHID project, grant EP/I011587/1.

References

Bezdek, J. C., and Hathaway, R. J. 2002. Some notes on alternating optimization. In *Advances in Soft Computing AFSS 2002*. Springer. 288–300.

Burt, C. M.; Piao, X.; Gaudi, F.; Busch, B.; and Taufik, N. 2008. Electric motor efficiency under variable frequencies and loads. *Journal of irrigation and drainage engineering* 134(2):129–136.

Chakraborty, P.; Marwah, M.; Arlitt, M. F.; and Ramakrishnan, N. 2012. Fine-grained photovoltaic output prediction using a bayesian ensemble. In *Association for the Advancement of Artificial Intelligence (AAAI) 2012*.

Chang, T. P. 2009. The suns apparent position and the optimal tilt angle of a solar collector in the northern hemisphere. *Solar Energy* 83(8):1274–1284.

Li, Z.; Liu, X.; and Tang, R. 2011. Optical performance of vertical single-axis tracked solar panels. *Renewable Energy* 36(1):64–68.

Liu, B., and Jordan, R. 1961. Daily insolation on surfaces tilted towards equator. *Solar Energy* 10:53.

Luque, A., and Hegedus, S. 2011. *Handbook of Photovoltaic Science and Engineering*. Wiley.

Lynch, W. A., and Salameh, Z. M. 1990. Simple electro-optically controlled dual-axis sun tracker. *Solar Energy* 45(2):65–69.

Mousazadeh, H.; Keyhani, A.; Javadi, A.; Mobli, H.; Abrinia, K.; and Sharifi, A. 2009. A review of principle and sun-tracking methods for maximizing solar systems output. *Renewable and Sustainable Energy Reviews* 13(8):1800–1818.

Myers, J. A. 1962. *Handbook of Equations for Mass and Area Properties of Various Geometrical Shapes*. NAVWEPS report 7827. U.S. Naval Ordnance Test Station.

Panagopoulos, A.; Chalkiadakis, G.; and Koutroulis, E. 2012. Predicting the power output of distributed renewable energy resources within a broad geographical region. In *European Conference on Artificial Intelligence (ECAI) 2012*, 981–986.

Peterka, J. A., and Derickson, R. G. 1992. Wind load design methods for ground-based heliostats and parabolic dish collectors.

Picard, A.; Davis, R. S.; Gläser, M.; and Fujii, K. 2008. Revised formula for the density of moist air (cipm-2007). *Metrologia* 45(2):149.

Ramchurn, S.; Vytelingum, P.; Rogers, A.; and Jennings, N. R. 2012. Putting the "smarts" into the smart grid: A grand challenge for artificial intelligence. *Communications of the ACM*.

Reda, I., and Andreas, A. 2004. Solar position algorithm for solar radiation applications. *Solar Energy* 76(5):577–589.

Roos, T. H. 2012. A wind loading correlation for an isolated square heliostat part 2: Moments and side forces.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. A Bradford book. Bradford Book.