

---

# Improving Solar Panel Efficiency Using Reinforcement Learning

---

**David Abel, Emily Reif, Michael L. Littman**

Department of Computer Science

Brown University

Providence, RI 02912

david.abel@brown.edu, emily\_reif@brown.edu, mlittman@cs.brown.edu

## Abstract

Solar panels sustainably harvest energy from the sun. To improve performance, panels are often equipped with a tracking mechanism that computes the sun's position in the sky throughout the day. Based on the tracker's estimate of the sun's location, a controller orients the panel to minimize the angle of incidence between solar radiant energy and the photovoltaic cells on the surface of the panel, increasing total energy harvested. Prior work has developed efficient tracking algorithms that accurately compute the sun's location to facilitate solar tracking and control. However, always pointing a panel directly at the sun does not account for diffuse irradiance in the sky, reflected irradiance from the ground and surrounding surfaces, or changing weather conditions (such as cloud coverage), all of which are contributing factors to the total energy harvested by a solar panel. In this work, we show that a reinforcement learning (RL) approach can increase the total energy harvested by solar panels by learning to dynamically account for such other factors. We advocate for the use of RL for solar panel control due to its *effectiveness*, *negligible cost*, and *versatility*. Our contribution is twofold: (1) an adaption of typical RL algorithms to the task of improving solar panel performance, and (2) an experimental validation in simulation based on typical solar and irradiance models for experimenting with solar panel control. We evaluate the utility of various RL approaches compared to an idealized controller, an efficient state-of-the-art direct tracking algorithm, and a fixed panel in our simulated environment. We experiment across different time scales, in different places on earth, and with dramatically different percepts (sun coordinates and raw images of the sky with and without clouds), consistently demonstrating that simple RL algorithms improve over existing baselines.

**Keywords:** Solar panels, Renewable Energy, Computational Sustainability

# 1 Introduction

Solar energy offers a pollution free and sustainable means of harvesting energy directly from the sun. Considerable effort has been directed toward maximizing the efficiency of end-to-end solar systems, including the design of photovoltaic cells or new photovoltaic materials [15] and solar tracking systems [2]. Solar tracking is especially important for maximizing performance of solar panels [6, 22, 12]. Given the proper sensors and hardware, tracking algorithms compute the relative location of the sun in the sky throughout the day. Then, a controller orients the panel to point at the sun, illustrated in Figure 1. The goal is to minimize the angle of incidence between incoming solar radiant energy and the grid of photovoltaic cells, as in Eke and Senturk [6] and Benghanem [1].

Prior work has consistently demonstrated that panels using a tracking system increase the total energy by a substantial amount: Eke and Senturk [6] report that a dual-axis tracker yielded 71 kW/h, compared to a fixed panel’s yield of 52 kW/h on the same day. They also report energy harvesting gains of dual-axis tracking systems over fixed systems varying from 15% to 40%, depending on the time of year. Mousazadeh *et al.* [18] report that gains from tracking vary between 0% and 100%, while Clifford and Eastwood [3] report a gain of 23% due to tracking in simulation. Solar tracking and control results in non-trivial benefits in solar photovoltaic systems.

Recent work in solar tracking has focused on algorithms that are sufficiently accurate to inform control of panels, building on the early work of Spencer [24]. The algorithm introduced by Reda and Andreas [21] computes the sun’s location in the sky within  $\pm 0.0003^\circ$  of accuracy, achieving the highest degree of accuracy of any known algorithm, but is computationally inefficient to the point of impracticality. Grena [8] overcomes these inefficiencies with a tracking algorithm that requires an order of magnitude fewer calculations while still achieving  $0.0027^\circ$  of accuracy.

However, prior literature suggests that a variety of factors contribute to the performance of a panel [12], and thus, pointing directly at the sun is not always optimal behavior. Specifically, the total solar irradiance falling on a panel is a combination of *direct*, *reflective*, and *diffuse* irradiance [1]. The diffuse irradiance typically varies between 15% and 55% of direct irradiance depending on factors like cloud coverage and the time of day [19], while a case study by the Cold Climate Housing Research Center in Fairbanks, Alaska reports reflective irradiance varying from 5% to 25% of direct irradiance [4]. The reflective irradiance depends on the percentage of irradiance reflected off the surrounding ground surface: Typical values for this percentage given by McEvoy *et al.* [17] vary between 17% (soil), 55% (concrete), and 90% (snow). Additionally, changing weather conditions can affect the optimal panel orientation [11]. Thus, optimal performance may involve prioritizing reflective or diffuse irradiance when direct sunlight is unavailable.

There are two additional shortcomings to the classical tracking approach. First, tracking algorithms take as input a variety of data that require additional hardware such as a barometer, thermometer, or GPS [9], increasing the total cost and system complexity. Second, tracking algorithms are only accurate for a fixed window of time: The algorithm of Grena [8] is noted as accurate until 2023 AD (due to the subtle movements of the earth and sun), while the algorithms in Grena [9] are reported as accurate until 2110 AD.

In this work, we advocate for the use of RL to optimize solar panel performance. A learned solar panel controller can account for weather change, cloud coverage, and diverse reflective indices of surroundings, offering an efficient yet adaptive solution that can optimize for the given availability of each type of solar irradiance without the need for complex hardware, regardless of the location or year. In this work, we advance solar panel control as an application area for RL, including a high fidelity simulation built using recently introduced models of solar irradiance, and validate of the utility of RL approaches for solar panel control.

## 1.1 Background

The amount of solar radiant energy contacting a surface on the earth’s surface (per unit area, per unit time) is called *irradiance* [7]. We denote the total irradiance hitting a panel as  $R_t$ , which, per the models developed by Kamali *et al.* [10], is approximated by the sum of the *direct* irradiance,  $R_d$ , *diffuse* irradiance (light from the sky),  $R_f$ , and *reflective* irradiance,  $R_r$  (reflected off the ground or other surfaces). Each of these components is modified by a scalar,  $\theta_d, \theta_f, \theta_r \in [0, 1]$ , denoting the effect of the angle of incidence between oncoming solar rays and the panel’s orientation, yielding the total:

$$R_t = R_d\theta_d + R_f\theta_f + R_r\theta_r \tag{1}$$

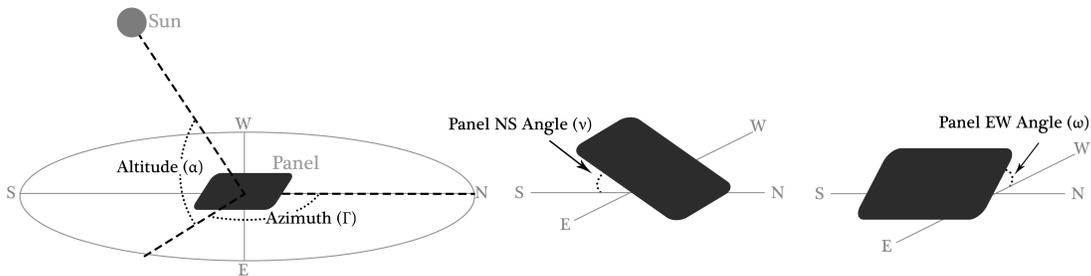


Figure 1: In the solar panel control problem, the panel changes orientation over time to maximize exposure to solar radiant energy.

Additionally, the components  $R_d$  and  $R_f$  are known to be effected by cloud coverage [13, 20, 26]. We attend to these details in describing our simulation in Section 2. A controller for a solar panel then seeks to maximize total irradiance,  $R_t$ , hitting the panel’s surface. In the case of solar trackers, a running assumption is that it is near optimal to orient the panel such that its normal vector is pointing at the sun, and thus arises the necessity for accurate solar tracking algorithms. There are many types of tracking methods, only a few of which we discuss in this work; for an in depth survey of solar tracking techniques, see Mousazadeh *et al.* [18].

## 2 Simulation

We introduce a high fidelity simulated environment to validate the use of RL for solar panel control. There are four basic stages to the simulation: (1) computing the sun’s location in the sky, relative to the panel, (2) computing  $R_d$ ,  $R_f$ , and  $R_r$ , (3) computing  $\theta_d$ ,  $\theta_f$ , and  $\theta_r$ , and (4) generating percepts.

**(1) Sun’s location in the sky:** For a given latitude, longitude, year, month, day, and time, we simulate the relative positions of the sun to the specified location on earth. Our simulation computes the sun’s altitude  $\alpha$  (angle: degrees above the horizon) and azimuth  $\Gamma$  (angle: clockwise degrees along the horizon relative to North) using the accurate tracker algorithm from Reda and Andreas [21] implemented in the open source library `pysolar`.<sup>1</sup> Due to space constraints, we do not give the full computation. Note, however, that a quicker approximation (also available in `pysolar`) is given by:

$$\alpha = \arcsin(\cos L \cos \delta \cos H + \sin L \sin \delta), \quad \Gamma = \arcsin\left(\frac{\cos \delta \sin H}{\cos \alpha}\right). \quad (2)$$

**(2) Computing  $R_d$ ,  $R_f$ ,  $R_r$ :** Given the sun’s altitude  $\alpha$  and azimuth  $\Gamma$ , we compute the  $R_d$ ,  $R_f$ , and  $R_r$  from the models of Threlkeld and Jordan [25] and Masters [16].

$$R_d = Ae^{-km}, \quad R_f = C \cdot R_d, \quad R_r = \rho R_d(\sin \alpha + C), \quad (3)$$

where  $A$  is the apparent extraterrestrial flux,  $k$  is the optical depth,  $m$  is the air mass ratio,  $\rho \in [0, 1]$  is a reflective index (albedo) denoting how reflective the ground is, and  $C$  is a sky diffusion factor, each given by the approximations:

$$m = \frac{1}{\sin \alpha}, \quad A = 1160 + \sin(0.99n - 271), \quad k = 0.174 + 0.035 \sin(0.99n - 99), \quad C = 0.095 + 0.04 \sin(0.99n - 99), \quad (4)$$

where  $n \in [1 : 365]$  is a day of the year.

**(3) Computing  $\theta_d$ ,  $\theta_f$ ,  $\theta_r$ :** Given the two angles describing the panel’s orientation ( $\nu$ : north-south tilt,  $\omega$ : east-west tilt), we simulate the amount of total irradiance actually hitting the panel’s surface. The models of Masters [16] compute  $\theta_t$  in terms of the cos-similarity between the panel’s normal vector,  $\vec{p}$ , and the sun’s vector,  $\vec{s}$ , (with the panel as the origin):

$$\vec{p} = [\sin(\nu) \cos(\omega) \quad \cos(\nu) \cos(\omega) \quad \cos(\nu) \cos(\omega)], \quad \vec{s} = [\sin(\pi - \Gamma) \cos(\alpha) \quad \cos(\pi - \Gamma) \cos(\alpha) \quad \sin(\alpha)]. \quad (5)$$

We then compute  $\theta_d$  as follows:

$$\theta_d = \frac{\vec{p} \cdot \vec{s}}{\|\vec{p}\| \|\vec{s}\|} = \sin(\nu) \cos(\omega) \sin(\pi - \Gamma) \cos(\alpha) + \cos(\nu) \cos(\omega) \cos(\pi - \Gamma) \cos(\alpha) + \cos(\nu) \cos(\omega) \sin(\alpha). \quad (6)$$

The diffuse irradiance incident angle  $\theta_f$  is given by a simple approximation—the solar collector is exposed to the fraction of the sky it points to—while  $\theta_r$  is given by the fraction of the ground the collector points to:

$$\theta_f = \frac{\cos \nu + \cos \omega}{2}, \quad \theta_r = \frac{2 - \cos \nu - \cos \omega}{2}. \quad (7)$$

**(4) Generating Percepts:** The final step in the simulation is to generate percepts (state variables) for the learning algorithms. In future work, we plan to control real solar panels with RL outside of simulation. For the real system we will equip each solar panel with a fish eye monocular camera to provide images of the sky as input for the RL algorithm. To approximate the real setting, our simulated environment supports percepts of three kinds:

1. The panel’s orientation and two angles representing the sun’s position in the sky relative to the panel (four state variables).
2. A  $16 \times 16$  synthesized grayscale image of the clear sky (256 state variables, each in the range  $[0, 1]$ ).
3. A  $16 \times 16$  synthesized grayscale image of the sky with simulated cloud cover (256 state variables, each in the range  $[0, 1]$ ).

For both image percepts, the perceived fraction of the sky changes as the panel moves to approximate having the camera mounted to the panel. In generating images, we ignore refractive effects on irradiance due to temperature and pressure.

Cloud cover is generated as Gaussian blobs in the synthesized images. The cloud conditions are randomized each morning at 4am, with the clouds moving from the left side of the image to the right (across the sky) throughout the course of the day (1 pixel per hour).

<sup>1</sup>[pysolar.org](http://pysolar.org)

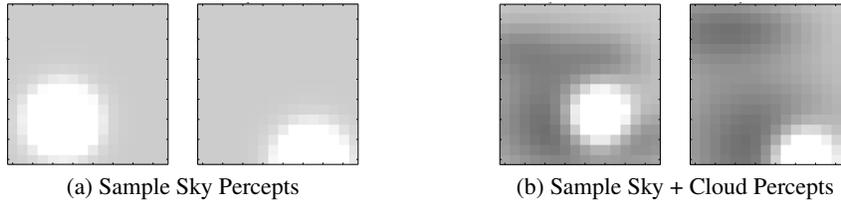


Figure 2: Example percepts given to the RL agent with no clouds (left) and simulated cloud coverage (right).

Each day at 4am, we generate between 1 and 5 clouds uniformly at random, with each cloud modeled as a Multivariate Gaussian. Parameters are randomized as:

$$\mu \sim [\text{Unif}(0 : N) \quad \text{Unif}(0 : N)], \quad \Sigma \sim \begin{bmatrix} \text{Unif}(2 : 6) & 0.2 \\ 0.2 & \text{Unif}(2 : 4) \end{bmatrix}. \quad (8)$$

Where  $\text{Unif}(x : y)$  denotes a uniform random sample from the natural interval  $[x : y]$ , and  $N$  is the dimension of the image (which we set to 16). The intensity of the cloud corresponds to the value of the Gaussian. Surprisingly, studies show that diffuse irradiance can be either magnified [23] or decreased [20] by cloud cover, depending on conditions. In light of these findings, our main interest in experimenting with clouds is to evaluate RL approaches with a rich state space, so we choose not to alter the diffuse irradiance. Direct irradiance ( $R_d$ ), however, is almost always decreased by cloud cover. Thus, if a cloud sits between the panel and the sun, we reduce the direct irradiance hitting the panel’s surface proportional to the cloud’s intensity.

### 3 Experiments

In each experiment, evaluation is done *online*, in that the agent is learning while acting to better parallel the nature of solar panel control. Our simulation is wrapped in an MDP, where the action space consists of five actions:  $\mathcal{A} = \{\text{tilt N}, \text{tilt E}, \text{tilt S}, \text{tilt W}, \text{nothing}\}$ . Executing the `nothing` action keeps the panel orientation fixed, while the other four each shift the panel  $2^\circ$  in their respective directions. Each decision step is equivalent to three minutes of time passing.

We experiment with two simple learning agents: *Q-Learning* and *SARSA*, each with a linear function approximator to estimate  $Q^*$ . To test the significance of modeling the sequential aspects of the problem, we also conduct experiments with *Q-Learning* with  $\gamma = 0$  (so it only maximizes immediate return) and *LinUCB* [14], a standard approach to non-sequential Contextual Bandits. We set exploration parameter  $\varepsilon_0 = 0.3$  and learning rate  $\eta_0 = 0.1$  with a standard annealing schedule, adopted from Darken and Moody [5]:

$$\eta_t = \frac{\eta_0}{(1.0 + \frac{t}{500})}, \quad \varepsilon_t = \frac{\varepsilon_0}{(1.0 + \frac{t}{500})}$$

Where  $t$  is a time step, and the update is performed every 500 time steps. For `q-lin` and `sarsa-lin` we set  $\gamma = 0.99$  to emphasize the long term consequences of behavior, contrasted with *LinUCB* and a short-sighted version of *Q-Learning* with  $\gamma = 0$  (`q-lin`,  $\gamma = 0$ ). Our benchmark algorithm is Algorithm 2 from Grena [9], an efficient but accurate solar tracking algorithm, coupled with a controller that always points perfectly at the tracker’s estimate of the sun’s location. We also provide results for a fixed panel to illustrate the importance of tracking (`fixed`), and a highly idealized controller that computes the perfect orientation at each decision step to illustrate an upper bound on possible performance (`optimal`). All of our code for running experiments and for reproducing results is publicly available.<sup>2</sup>

Figure 3 illustrates the cumulative irradiance exposed to each panel in the Australia experiments (top) and the Iceland experiments (bottom). Notably, with the simple percept of the true sun angles, *all* of the learning algorithms outperform the baseline tracker and fixed panel in Australia, while in Iceland, `lin-ucb` performs worse than the `fixed` panel for the first two percepts. When just the image is provided in Australia, `lin-ucb` achieves by far the best performance; we hypothesize that this is due to *LinUCB*’s informed approach to exploration compared to the  $\varepsilon$ -greedy used by `sarsa-lin` and both `q-lin` and `q-lin`,  $\gamma = 0$ . Additionally, when just the sun is in the image, there is little incentive to forecast expected future reward beyond the immediate next step, which may explain the success of `lin-ucb`. This is further corroborated by the fact that `q-lin`,  $\gamma = 0$  does quite well in the experiment as well. Conversely, we see that `lin-ucb` continues to struggle in Iceland. When clouds are present, `lin-ucb` performs comparably to the other learners. The cloudy image percepts pose a challenging RL problem, but still we see that the simple approaches achieve similar performance as the `grena-tracker`, and note the substantial room for improvement from further training or more complex learners. In Iceland, the results are largely the same as Australia, though we note that the `grena-tracker` does better.

In all cases, we note that there is room for improvement, suggesting that more sophisticated approaches may have more success on real panels than current techniques. In all our simulations, the `grena-tracker` consistently outperforms the `fixed-panel` by around 25%, consistent with previously published results [6, 18, 3].

<sup>2</sup>[https://github.com/david-abel/solar\\_panels\\_rl/experiments](https://github.com/david-abel/solar_panels_rl/experiments)

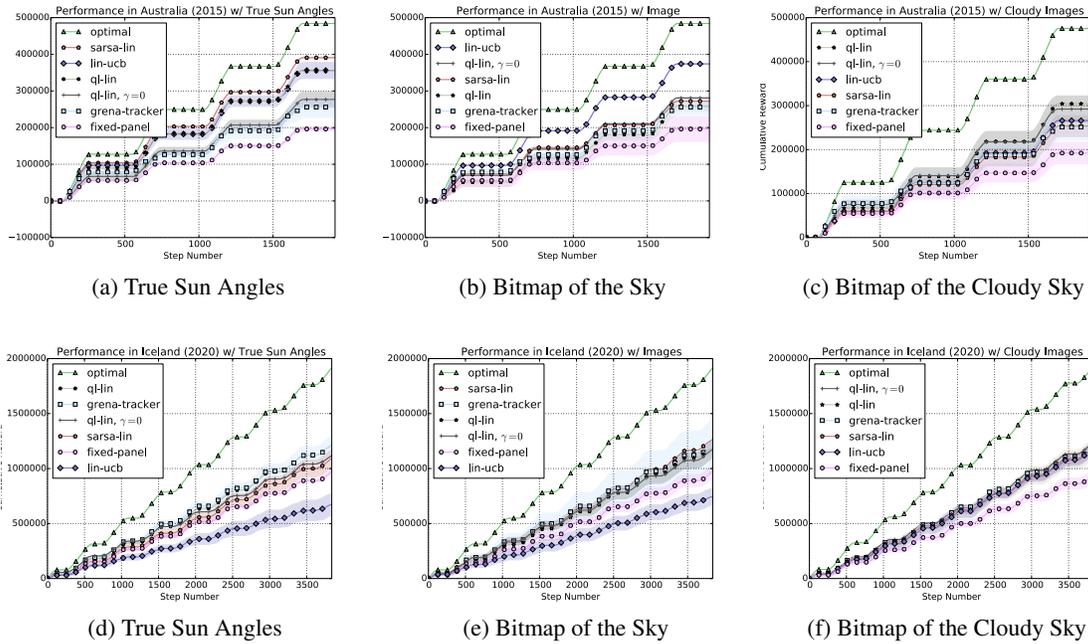


Figure 3: Cumulative irradiance falling on the panel's surface given different percepts over four days in Mildura, Australia in July of 2015 (top) and eight days in Reykjavik, Iceland in July of 2020 (bottom).

## References

- [1] M. Benganem. Optimization of tilt angle for solar panel: Case study for Madinah, Saudi Arabia. *Applied Energy*, 88(4):1427–1433, 2011.
- [2] Eduardo F Camacho and Manuel Berenguel. Control of solar energy systems1. *IFAC Proceedings Volumes*, 45(15):848–855, 2012.
- [3] MJ Clifford and D Eastwood. Design of a novel passive solar tracker. *Solar Energy*, 77(3):269–280, 2004.
- [4] Ryan Colgan, Nathan Wiltse, Michael Lilly, Ben LaRue, and Greg Egan. Performance of photovoltaic arrays. *Cold Climate Housing Research Center*, 2010.
- [5] Christian Darken and John E Moody. Note on learning rate schedules for stochastic optimization. In *NIPS*, volume 91, pages 832–838, 1990.
- [6] Rustu Eke and Ali Senturk. Performance comparison of a double-axis sun tracking versus fixed PV system. *Solar Energy*, 86(9):2665–2672, 2012.
- [7] D Yogi Goswami, Frank Kreith, and Jan F Kreider. *Principles of solar engineering*. CRC Press, 2000.
- [8] Roberto Grena. An algorithm for the computation of the solar position. *Solar Energy*, 82(5):462–470, 2008.
- [9] Roberto Grena. Five new algorithms for the computation of sun position from 2010 to 2110. *Solar Energy*, 86(5):1323–1337, 2012.
- [10] Gh A Kamali, I Moradi, and A Khalili. Estimating solar radiation on tilted surfaces with various orientations: a study case in karaj (iran). *Theoretical and applied climatology*, 84(4):235–241, 2006.
- [11] Nelson A. Kelly and Thomas L. Gibson. Improved photovoltaic energy output for cloudy conditions with a solar tracking system. *Solar Energy*, 83(11):2092–2102, 2009.
- [12] D L King, William E Boyson, and J A Kratochvil. Analysis of factors influencing the annual energy production of photovoltaic systems. *Conference Record of the Twenty-Ninth IEEE Photovoltaic Specialists Conference, 2002.*, pages 1356–1361, 2001.
- [13] Danny HW Li, Chris CS Lau, and Joseph C Lam. Overcast sky conditions and luminance distribution in hong kong. *Building and Environment*, 39(1):101–108, 2004.
- [14] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670. ACM, 2010.
- [15] Yongfang Li. Molecular design of photovoltaic materials for polymer solar cells: toward suitable electronic energy levels and broad absorption. *Accounts of Chemical Research*, 45(5):723–733, 2012.
- [16] Gilbert M Masters. *Renewable and efficient electric power systems*. John Wiley & Sons, 2013.
- [17] Augustin McEvoy, Tom Markvart, Luis Castañer, T Markvart, and Luis Castaner. *Practical handbook of photovoltaics: fundamentals and applications*. Elsevier, 2003.
- [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, 2009.
- [19] William A Peterson and Inge Dirmhirn. The ratio of diffuse to direct solar irradiance (perpendicular to the sun's rays) with clear skies a conserved quantity throughout the day. *Journal of Applied Meteorology*, 20(7):826–828, 1981.
- [20] G Pfister, RL McKenzie, JB Liley, A Thomas, BW Forgan, and Charles N Long. Cloud coverage based on all-sky imaging and its impact on surface solar irradiance. *Journal of Applied Meteorology*, 42(10):1421–1434, 2003.
- [21] Ibrahim Reda and Afshin Andreas. Solar position algorithm for solar radiation applications. *Solar energy*, 76(5):577–589, 2004.
- [22] J Rizk and Y Chaiko. Solar Tracking System: More Efficient Use of Solar Panels. *Proceedings of World Academy of Science: Engineering & Technology*, 43:313–315, 2008.
- [23] Nathan Robinson. *Solar radiation*. Elsevier, 1966.
- [24] JW Spencer. Fourier series representation of the position of the sun. *Search*, 2(5):172–172, 1971.
- [25] JL Threlkeld and RC Jordan. Direct solar radiation available on clear days. *Heat., Piping Air Cond.*, 29(12), 1957.
- [26] P Tzoumanikas, E Nikitidou, AF Bais, and A Kazantzidis. The effect of clouds on surface solar irradiance, based on data from an all-sky imaging system. *Renewable Energy*, 95:314–322, 2016.