SunShade: Enabling Software-defined Solar-Powered Systems

Akansha Singh
UMass Amherst
akanshasingh@umass.edu

Stephen Lee
UMass Amherst
stephenlee@cs.umass.edu

David Irwin
UMass Amherst
deirwin@umass.edu

Prashant Shenoy
UMass Amherst
shenoy@cs.umass.edu

ABSTRACT

The electric grid was not designed to support the large-scale penetration of intermittent solar generation. As a result, current policies place hard caps on the solar capacity that may connect to the grid. Unfortunately, users are increasingly hitting these caps, which is restricting the natural growth of solar power. To address the problem, we propose Software-defined Solar-powered (SDS) systems that dynamically regulate the amount of solar power that flows into the grid. To enable SDS systems, this paper introduces fundamental mechanisms for programmatically controlling the size of solar flows, including mechanisms to both enforce an absolute limit on solar output and a new class of Weighted Power Point Tracking (WPPT) algorithms that enforce a relative limit on solar output as a fraction of its maximum power point (MPP). We implement an SDS prototype, called SunShade, and evaluate tradeoffs in the accuracy and fidelity of these mechanisms to enforce limits on solar flows. For example, we quantify the effects of variable conditions, such as clouds, passersby, and other shading, on the fidelity of a search-based WPPT algorithm, which must periodically deviate from its cap to discover changes in the MPP that affect the cap's accuracy.

CCS CONCEPTS

•Applied computing →Computer-aided design; •Computer systems organization →Sensors and actuators; •Hardware →Power networks;

KEYWORDS

Software-defined Solar, Weighted Power Point Tracking

ACM Reference format:
Akansha Singh, Stephen Lee, David Irwin, and Prashant Shenoy. 2017. SunShade: Enabling Software-defined Solar-Powered Systems. In *Proceedings of The 8th ACM/IEEE International Conference on Cyber-Physical Systems, Pittsburgh, PA USA, April 2017 (ICCPS 2017),* 10 pages.
DOI: http://dx.doi.org/10.1145/3055004.3055013

ICCPS 2017, Pittsburgh, PA USA

978-1-4503-4965-9/17/04…$$15.00
DOI: http://dx.doi.org/10.1145/3055004.3055013

1 INTRODUCTION

The electric grid is in the midst of a profound transformation, as users are increasingly generating their own energy locally from renewable sources rather than purchasing it from electric utilities. This transformation is being spurred by exponential decreases in the cost of solar modules, which have fallen 60% since 2011 [1]. Rapidly falling prices have in turn driven significant increases in distributed solar generation. Nearly all solar deployments are "grid-tied," enabling them to draw power from the grid when their local demand exceeds solar generation and feed power into the grid when their local solar generation exceeds demand. However, these grid-tied systems impose a burden on the grid to absorb a building's energy surpluses and make up for its energy deficits.

The electric grid was not designed to support such decentralized and intermittent energy generation by millions of individual users. Instead, the grid imposes a rigid top-down hierarchy where large, highly-regulated utilities generate (and purchase) energy to meet the demand of their customers and maintain grid stability. To do so, utilities continuously balance electricity's supply and demand in real time by regulating generator power output. Since the energy demand (or "load") profile of individual users is stochastic, such real-time balancing is only possible because the sum of load profiles across many users tends to be smooth and highly predictable. As a result, utilities can plan when to activate (or "dispatch") generators in advance to satisfy large increases in demand.

Distributed solar generation at large scales fundamentally alters this paradigm by increasing the stochasticity of user load profiles, even when aggregating them. While solar power output can change instantaneously, e.g., due to passing clouds, dispatchable generators are mechanical devices that take some time to activate and adjust their power output, which prevents them from maintaining high power quality when compensating for rapid solar variations. In general, the grid faces significant operational challenges when renewable penetration approaches 10% [14], necessitating additional energy storage or sophisticated demand-side load management.

Due to the challenges above, states generally place hard limits ("caps") on the collective solar capacity that may connect to the grid. However, due to the rapid growth in solar power, users are now starting to hit these caps. For example, Massachusetts reached its cap in the summer of 2015, which immediately halted construction on 134MW of new solar deployments. The legislature did not pass a stop-gap bill to raise the cap until April 2016 [13]. In Hawaii, where 12% of residents have rooftop solar, utilities barred additional residents from installing grid-tied solar for two years until the

government recently intervened [3]. Similar tensions now exist in Germany [19], Australia [12], and Italy [4].

The underlying reason for the caps above is that the grid exercises no control over when and how much solar power flows into it, enabling unlimited solar power to flow in from any connected solar-powered system, even when it might compromise grid reliability and power quality. To address the problem, we propose Software-defined Solar-powered (SDS) systems capable of dynamically regulating the power they let "flow" into the grid, similar to how the Internet and other networks strictly regulate data transmission. By adaptively controlling the size of *solar power flows*, SDS systems provide the grid the necessary tools to control solar power in real-time to balance supply and demand, thereby reducing the need for artificial regulatory caps on solar connections. The current approach to allocating the grid's available solar capacity is essentially a static peak-based first-come-first-serve policy that only approves new solar connections if the grid can handle *all* solar connections operating at their peak capacity. This approach is highly inefficient, and wastes much of the grid's potential to transmit solar power.

The goal of SDS systems is to enable all users to freely connect to the grid and dynamically share its capacity to transmit solar power. This principle represents a form of *grid neutrality*, akin to net neutrality, where the grid treats all solar energy contributions equally without discriminating between users. Just as in the Internet, the rate at which users inject energy should be dynamically regulated to maximize the grid's available solar capacity (i.e., maximize goodput), maintain the supply/demand balance (i.e., prevent congestion collapse), and fairly share the capacity among connected users. Of course, controlling solar power differs in key ways from regulating data transmission. Since data transmission is packet-based, adjusting sending rates is simple and only requires regulating the time between packet transmissions. In contrast, solar power is continuous and thus requires different mechanisms for control. In addition, the maximum "sending rate" of a solar power flow varies continuously in real time based on physical properties, e.g., the weather and the sun, and is not a function of an arbitrary application's demands.

To provide a foundation for SDS systems, this paper proposes fundamental software mechanisms to control the "sending rate" of solar flows. Our first SDS mechanism enables software to directly control solar flows by placing an arbitrary *absolute cap* on solar power generation, akin to a network bandwidth cap. However, as we discuss, controlling solar flows using absolute caps may unfairly penalize large, or optimally configured, solar installations, e.g., by restricting the percentage of their maximum power that can flow into the grid. Thus, we introduce a new class of *Weighted Power Point Tracking* (WPPT) algorithms, akin to weighted proportional-share allocation in networking, that enforce a relative limit on solar output as a fraction of its (dynamically changing) maximum power point. We implement these mechanisms in a prototype SDS system we have developed, called SunShade. In designing SunShade, we make the following contributions.

Software-defined Solar-powered Systems. We introduce Software-defined Solar-powered (SDS) systems to enable innovation in the design of higher-level solar transmission protocols to support arbitrarily high solar penetrations in the grid. We contrast SDS systems with existing work on smart inverters and active solar curtailment, which enable specific operational modes and not software programmability.

Solar Flow Control Mechanisms. We design two fundamental mechanisms for enabling software to control solar flow rates, inspired by similar mechanisms in networking and operating systems. Absolute capping enforces hard caps on solar output, while WPPT enforces a relative cap based on a system's changing maximum power point. We define two WPPT variants—model-based and search-based—and examine their tradeoffs in complexity, accuracy, and performance.

Implementation and Evaluation. We implement a prototype SDS system, called SunShade, and evaluate the fidelity of the mechanisms above. In particular, we quantify the effects of variable conditions, such as clouds, passersby, and other shading, on the fidelity of a search-based WPPT algorithm, which must periodically deviate from its cap to discover changes in the MPP that affect the cap's accuracy.

2 OVERVIEW

Solar-powered buildings and homes typically connect to the grid, enabling them to feed power into the grid and draw power from the grid as solar generation fluctuates. Such grid-tied systems are much more efficient (and less expensive) than "off grid" installations because they reduce (or eliminate) the need for local energy storage, which is expensive to install and maintain. Grid-tied systems are also inherently more efficient because they enable buildings to exchange power to balance demand, i.e., by permitting one building to consume surplus solar power from a neighbor. However, injecting arbitrarily large amounts of intermittent solar energy into the grid is problematic, as utilities must offset any fluctuations in solar generation to maintain grid balance and power quality, e.g., by generating less power during solar surpluses and more power during solar deficits. Unfortunately, installing and maintaining enough energy storage capacity at grid scales to smooth fluctuations remains prohibitively expensive at high solar penetration levels, e.g., >10% generation from solar.

As a result, governments strictly regulate grid solar connections, typically by setting a hard limit on the maximum solar capacity that may connect to the grid. These limits—set statically through legislation—are generally low. In Massachusetts, the cap that was hit in summer 2015 was set at 4% of peak grid load [13]. Admission control policies for solar installations based on static peak-based capping are highly wasteful because they prevent fully utilizing the grid's capacity to transmit solar power. To understand why, consider that solar-powered systems are rarely generating at peak power due to non-optimal operating conditions, e.g., from clouds, shade from obstructions, early morning or evening hours, non-optimal tilts/orientations, seasonal variations, etc. In fact, solar installations are capable of generating their peak power at only a single instant over the entire year: at solar noon on the summer solstice, assuming clear skies. Thus, generating the same amount of solar power on cloudy days or in the winter requires energy contributions from more solar-powered systems than on sunny summer days. Unfortunately, static peak-based caps prevent new

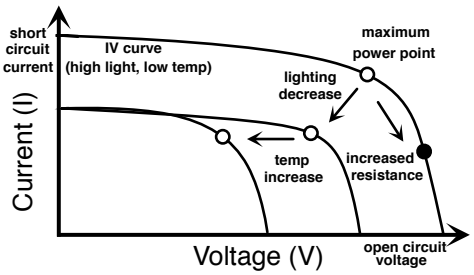


Figure 1: I-V curve for a typical solar module, and the effect of changes in lighting and temperature.

solar connections unless the grid can safely service all systems operating at their peak.

Any approach to dynamically regulating solar output must ensure *fairness* between solar systems. We define an allocation of solar flow rates as fair if over some time period τ, the solar flows are able to contribute the same percentage of their maximum possible generation potential to the grid. This definition of fairness normalizes for the size of the deployment, and also incentivizes an optimal configuration. For example, physical characteristics, such as non-optimal tilts and orientations and shade from surrounding buildings, lower a deployment's maximum generation potential and thus its fair allocation. Unfair allocations of solar flows are undesirable because they reduce the compensation users receive for the solar energy they contribute to the grid, and increase users' local energy storage requirements (to store the energy they cannot contribute). In this paper, we focus narrowly on introducing new mechanisms to enforce setting proportional flow rates in SDS systems. Using these mechanisms to globally and dynamically regulate solar flows across multiple SDS systems to ensure fairness is outside of our scope.

2.1 Background

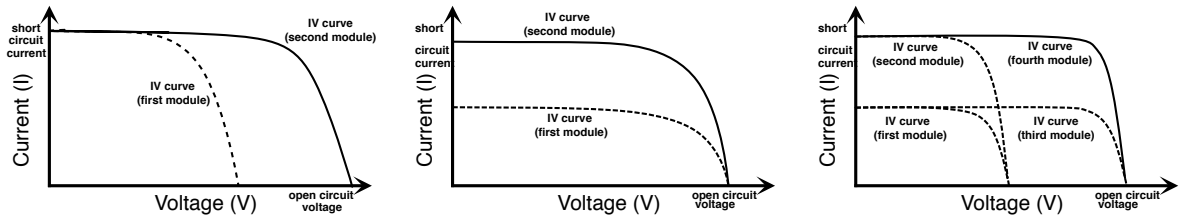
SDS systems enable software to *control* the size of solar power flows into the grid. Modern solar-powered systems already actively control solar power output within their *inverter*, which converts the DC electricity generated by the solar modules into AC electricity that is synchronized with the grid's AC electricity, e.g., to the same frequency and phase. Inverters typically implement an embedded algorithm for Maximum Power Point Tracking (MPPT) that constantly adjusts the deployment's operating voltage to maximize its power generation, as the current produced by solar modules varies non-linearly with voltage. However, existing solar inverters generally do not expose such control mechanisms to higher-level software. As we discuss, new smart inverters support other operating modes that implement embedded control algorithms beyond MPPT, e.g., VAR control, voltage/frequency ride-through, etc., but do not permit programmatic control by higher layers. In contrast, we focus on exposing programmatic interfaces to leverage similar inverter control mechanisms for regulating solar power output.

The primary factor that affects a solar deployment's maximum possible production is its solar insolation, i.e. the amount of solar radiation that is incident on the solar modules' area. The amount of solar insolation is affected by numerous variables, including the weather, angle of the sun in the sky (which varies across the day and year), shade from neighboring buildings and trees, modules' tilt and orientation, etc. Given these factors, a typical solar module is capable of operating at a range of different current and voltage levels, which govern its actual power output. The operating region of a solar system is governed by its *I-V curve*, as depicted in Figure 1. The figure shows a solar module's output current across a range of voltages (as dictated by the applied resistance), where the solar power output is simply the product of the voltage and current. Due to the nature of the I-V curve, the solar output power *changes* at different operating voltages. Specifically, since the I-V curve is initially flat, as the operating voltage increases, the output current remains virtually unchanged, leading to an *increase* in power output. However, after reaching the knee of the curve, any further increase in operating voltage yields a corresponding reduction in current, and hence the power begins to drop. Thus, the solar output rises with increasing voltage up to a point and then precipitously drops. As a result, each I-V curve has an optimal operating voltage V_{opt} that maximizes its output.

Note that the precise shape of the I-V curve is dynamic and changes continuously. For example, the maximum power point decreases as the solar insolation decreases, causing the curve to contract along both the x-axis and y-axis as depicted. In addition, the solar cell temperature also affects the shape of the curve, expanding and contracting it along the x-axis. While Figure 1 depicts an idealized curve for a single solar module, solar systems are typically composed of multiple modules wired (or "strung") together and connected to a single inverter. In this case, the I-V curve of the aggregate solar circuit is a combination of the I-V curves of each module. Figure 2 shows how the combined I-V curve is a composition of each module's I-V curve when wiring modules in series (a), in parallel (b), and a combination of the two (c). In particular, two modules wired in series operate at the same current, but have additive voltage, while two modules wired in parallel operate at the same voltage, but have additive current. The characteristics of each module may then change independently, affecting both the output of the other modules and system's aggregate I-V curve. For example, two connected modules may be installed with different tilts at different orientations, causing a shadow to cover one but not the other. If wired in series, the module producing the lowest current restricts the current generated by the other modules, reducing the entire array's output.

MPPT algorithms dynamically adjust the system's voltage to maximize power generation by operating at the "knee" of the I-V curve as the curve changes. Inverters implement MPPT algorithms using a DC-to-DC buck-boost converter that is able to adjust output voltage to be greater than or less than input voltage. Buck-boost converters typically use pulse-width modulation (PWM) to vary their duty cycle, which also varies the input/output voltage. There is a large body of prior work on developing Maximum Power Point Tracking (MPPT) algorithms—examples include the perturb and observe, current sweep, incremental conductance, and constant voltage ratio algorithms amongst many others. MPPT algorithm design is well-studied and presents many tradeoffs in optimizing power accuracy, convergence speed, implementation complexity, initialization procedures, etc. [2].

The most common MPPT algorithm is the Perturb and Observe (P&O) algorithm. This algorithm perturbs the voltage by a small amount, and then measures the instantaneous current and voltage



(a) Array in Series (b) Array in Parallel (c) Array Series-Parallel

Figure 2: Example I-V curve for a solar array formed from wiring multiple solar modules in series (a), in parallel (b), and a combination of the two (c).

to calculate the new power (P_t) and compares it to the power P_{t-1} at the previous voltage. If the change in power is positive, it continues to perturb the voltage in the same direction; if the change is negative then it reverses the direction of its search. Simple P&O algorithms use a fixed voltage step size on each iteration, while more sophisticated variations adapt the step size, e.g., proportional to the slope of the P-V curve $\frac{\Delta P}{\Delta V}$, to converge more quickly.

2.2 Controlling Solar Flows

SDS-enabled inverters use the same basic functions as MPPT to regulate solar power flows by operating at points other than the maximum power point (MPP). For example, an inverter could decrease the output below the MPP, or after decreasing output, could then increase output back to the MPP. By controlling the operating voltage, an inverter can precisely control solar output up to its MPP. Note that inverters may be used in conjunction with solar charge controllers, enabling them to either dissipate excess power when operating below the MPP (if there is no battery) or store the excess power in a battery. In the latter case, the solar charge controller circuitry implements the MPPT algorithm.

Interestingly, while solar power is characterized as inherently intermittent, only its MPP is intermittent. Below the variable MPP, solar inverters are capable of making rapid and precise changes in solar output, and generally have more flexibility to rapidly and precisely control power output than mechanical generators, which have physical limitations on the speed at which they can increase or decrease their revolutions. As we discuss in §6, active control of solar power to support increased solar penetration is an emerging area. However, prior work focuses largely on coarse solar curtailment during times of peak generation (akin to coarse demand response), and not on exposing mechanisms for fair, fine-grained regulation of solar output to higher-level software.

3 SDS CONTROL MECHANISMS

We design two useful mechanisms to control solar flows. Our first mechanism enforces an *absolute cap* on solar output, wherein the solar output is capped based on a specified limit. Formally, this cap is specified by a tuple (P_{cap}, t), which imposes an upper limit P_{cap} on the power for a duration t. Our second mechanism is a class of Weighted Power Point Tracking algorithms that enforce relative caps, such that power output is capped as a percentage of the system's time-varying maximum output. Formally, this cap is specified by a tuple (Δ, t) which indicates that power should be limited to a fraction Δ of the maximum power over the duration t, where $0 < \Delta \leq 1$. Note that, since the maximum power output is constantly changing, the absolute power generated from the relative cap also changes. While absolute capping enforces a strict power limit, WPPT enforces a "fair" limit across deployments with different characteristics.

Our SunShade prototype exposes a narrow interface that enables software to set and alter either the absolute cap P_{max} or the weight Δ. Note that this interface *does not* expose direct programmatic control of the voltage, but rather internally determines the appropriate operating voltage to enforce the specified power output. This is akin to software-defined networks that expose forwarding mechanisms to an external controller in the control plane, but do not expose direct control of the data plane's packet processing. Exposing direct control of voltage lowers the barrier to introducing deviant behavior into the grid, and could enable sophisticated grid attacks. In the past, grid interconnection standards have prevented inverters from actively adjusting their power output outside of using MPPT (see IEEE 1547-2003 [6]). However, these standards are changing to permit the basic control functions we propose (IEEE 1547a-2014 [7]), as solar capacity increases and smart inverters become more commonplace.

3.1 Absolute Power Capping

As discussed in the previous section, the Perturb and Observe (P&O) algorithm is the most widely used algorithm for tracking the MPP. To support absolute power capping, we adapt the classic P&O algorithm to ensure the solar output operates at or below a specified power limit. Algorithm 1 shows the pseudo-code for setting an absolute power cap on solar output. Similar to the P&O algorithm, the algorithm uses the instantaneous voltage and current to calculate the current power P_t. Power P_t is then compared to the previous power P_{t-1} to determine if there has been a change in power.

To operate at a given power cap P_{cap}, the algorithm simply compares P_{cap} to the current power P_t. If P_t is less or more than P_{cap}, the voltage is perturbed to increase or decrease, respectively, the power. To increase and decrease the magnitude of power, the instantaneous power P_t is compared to previous power P_{t-1}. If the change in power is positive, the algorithm continues to perturb in the same direction, else the direction of the perturbation is reversed. Upon reaching the limit P_{cap}, the instantaneous power output then oscillates around the limit. In this case, a bigger step size results in larger oscillations around the limit, but faster convergence, while a smaller step size has smaller oscillations, but has a slower convergence. As a result, a larger step size is more appropriate in scenarios where conditions are changing rapidly, while a smaller step size is more appropriate under stable conditions. While this

Algorithm 1 Absolute capping via modified P&O algorithm.

```
if Pcap != P
    if Pcap > P
        %increase power
        if P > Pold
            if V > Vold
                D = Dold - deltaD;
            else
                D = Dold + deltaD;
            end
        else
            if V > Vold
                D = Dold + deltaD;
            else
                D = Dold - deltaD;
            end
        end
    else
        %decrease power
        if P > Pold
            if V > Vold
                D = Dold + deltaD;
            else
                D = Dold - deltaD;
            end
        else
            if V > Vold
                D = Dold - deltaD;
            else
                D = Dold + deltaD;
            end
        end
    end
else D=Dold;
end
```

algorithm uses a fixed voltage step size, we can also use an adaptive voltage step size proportional to the difference between P_{cap} and P_t to converge faster after large variations. The only external input signal required is the absolute power cap P_{cap}. If the current maximum power point P_{mpp} is higher than the allowed limit, the system adjusts to operate at P_{cap}, else it operates at P_{mpp}.

3.2 Weighted Power Point Tracking

Weighted Power Point Tracking (WPPT) caps the power output such that it maintains output at a fixed fraction of the maximum power point P_{mpp}. Software sets the weight Δ between 0 and 1. To strictly enforce a weighted cap at any given time, the system must know the maximum power point P_{mpp} to compute the appropriate weighted power point. Thus, in this case, the absolute power cap is $P_{cap} = \Delta * P_{mpp}$ and changes dynamically over time. The challenge with WPPT is that the maximum power point is not well-known, and there is a cost to finding it. We describe two approaches to determine the P_{mpp} below. After determining P_{mpp}, we compute the weighted cap ΔP_{mpp} and use our absolute capping P&O algorithm above to maintain the weighted cap.

3.2.1 Search-based Approach. The search-based approach operates by periodically computing the actual maximum power point P_{mpp} using the standard P&O algorithm (or any other MPPT algorithm). The fidelity of the search-based approach in tracking the true weighted power point is a function of the variability in the MPP, the convergence speed of the MPPT algorithm, and the search interval. For example, if MPP variability is high and we search infrequently, then the fidelity will be low, as our weighted cap will not accurately reflect the actual cap (which is changing as a function of the MPP). In contrast, if variability is low, and we search frequently, then we will deviate from the weighted cap in

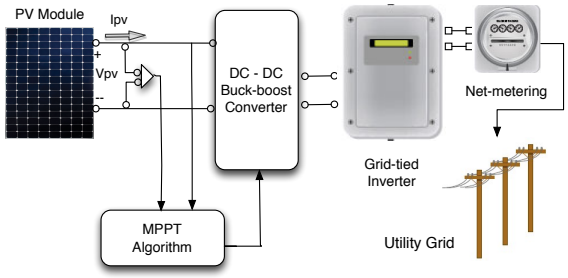


Figure 3: Depiction of our SunShade simulator.

searching, and thus generate more power than the cap dictates, and introduce spikes in the system's power output.

Our search-based approach invokes the P&O algorithm at a configurable interval with a specified search duration until the algorithm oscillates for multiple iterations around the maximum power point. The algorithm then halts and returns the P_{mpp}. The algorithm then uses the absolute capping algorithm to find the voltage V_{cap} that results in an absolute power cap of ΔP_{mpp}. This voltage will change based on weather conditions and time. The search-based approach may observe the changes in power at V_{cap} to determine when the MPP has deviated from previous MPP and trigger a new search. Our implementation supports setting a fixed search interval or an adaptive search interval that triggers a new search once V_{cap} has deviated by a configurable threshold.

3.2.2 Model-based Approach. The search-based approach has accuracy limitations because it must deviate from the true weighted cap to find the current MPP. These deviations may reduce accuracy to unacceptable levels under highly variable conditions that requires frequent searching. An alternative approach is to compute the MPP based on a model of the solar system's maximum output at any given time. This model may be constructed either empirically based on data collected by the inverter and weather sensors, or analytically given specifications of the solar panels, including their type, tilt, orientation, wiring, etc.

Building an empirical model has the advantage of not requiring a priori knowledge of the deployment, since the system behavior is learned from empirical observations. However, empirical models take time to build, as they require collecting data on the maximum power point under many different environmental conditions. To build such an empirical model, the inverter would use conventional MPPT to operate at its maximum power point for a long period of time to collect current and voltage values under many different ambient conditions with different temperatures and solar radiation levels. While the temperatures and solar radiation levels could be estimated from a local weather station, e.g., via Weather Underground, it is more accurate to link the inverter with external temperature and solar radiation sensors to record actual conditions. After recording the current and voltage at the maximum power point for many different values of temperature, solar radiation, and time, it can use standard techniques to build a model that predicts the current, voltage, and maximum power point for any values of temperature, solar radiation, and time [8].

The amount of time and data an inverter must collect to build an accurate model varies with each location. For example, in San

Diego, CA, where the climate is nearly constant year-round at 24°C and sunny, an accurate model will take little time to build, while in a highly variable climate, such as in the Northeast, it might take an entire year. As a result, we focus on an approach that analytically models a solar deployment based on its specific characteristics. While this approach requires configuring the inverter with details specific to each deployment, it requires no extended period of operation to collect training data. In the model-based approach, we are given the deployment specifications that dictate a model of the I-V curve for the deployment based on solar radiation and temperature. Again, we assume that the inverter uses external sensors to measure solar radiation, e.g., using a pyranometer, and temperature at the location. The model then infers the MPP based on the radiation and temperature levels. Note that no model is perfect; thus, the fidelity of this approach is ultimately a function of the model's accuracy. Another drawback of the model-based approach is that it requires irradiance and temperature sensors, which add to the cost and complexity of a solar deployment.

4 IMPLEMENTATION

We implement our SDS rate control mechanisms in simulation and in a small-scale SunShade prototype. Our simulation leverages Matlab's Simulink library (SimPowerSystems) for simulating a solar deployment's output based on its electrical characteristics, irradiance, and temperature.

Matlab includes a flexible solar cell model that we configure to match the panel from our small-scale prototype, and an implementation of the P&O MPPT algorithm that tracks maximum solar output as a function of solar radiation and temperature. The model shows a close, albeit imperfect, fit to the published data provided by the manufacturer. In order to compute the power cap P_{mpp}, our algorithm measures irradiance and temperature from the sensors, which it provides as input to the model. Our simulator implements absolute power capping and the two WPPT variants from the previous section by modifying the existing P&O algorithm in Matlab. Our simulator takes as input, data traces gathered from a pyranometer that measures solar radiation, and a temperature sensor. We are also able to generate synthetic traces of solar radiation and temperature for the simulator to test the fidelity of the mechanisms above under arbitrary conditions. Note that Matlab simulations are considered highly accurate and frequently used as the only means of evaluating new MPPT algorithms in the power systems community. Figure 3 depicts the circuit diagram of our SunShade simulator.

In addition to our simulations, we also construct a SunShade prototype to evaluate its mechanisms under realistic conditions. For our prototype, we connect voltage and current sensors between a load and a small solar panel (rated at a ideal peak capacity of 25W), which independently measure voltage and current. Rather than employ an embedded buck-boost converter, we use a programmable load—the BK Precision 8500 Programmable Load—to control the panel's operating voltage. Using the programmable load enables rapid experimentation by allowing us control operating voltage remotely from a server via `python`, rather than embedding such control into the buck-boost converter's firmware. Note that the programmable load is functionally equivalent to the buck-boost converter, and uses the same PWM mechanism to vary the panel's operating voltage. The primary difference is that the minimum reaction time—the time between two changes in voltage—on the programmable load is ~100ms due to the latency imposed by the serial connection. We could reduce this latency to near that provided by a typical buck-boost converter, e.g., tens of milliseconds, by using a modern I/O interface, such as USB. Figure 4 shows current and voltage from our SunShade prototype that reflect the expansion and contraction of the I-V curve, similar to Figure 1 from §2, as the intensity of light changes over the day.

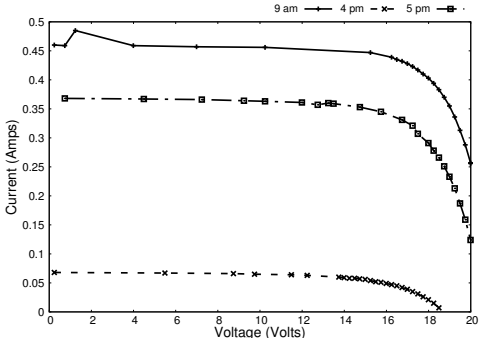


Figure 4: Expansion and contraction of the I-V curve for our SunShade prototype, as light intensity changes throughout the day.

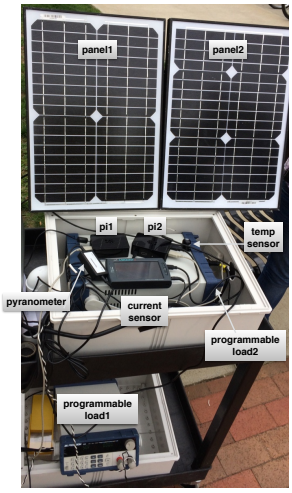


Figure 5: Depiction of our SunShade prototype.

Finally, evaluating WPPT's fidelity requires comparing its results to the actual weighted power point dictated by the real MPP. To support such comparisons, we construct an additional parallel prototype to run MPPT that uses the same solar panel as SunShade. We then place the two systems directly adjacent to each other so they are subject to nearly identical solar conditions. Figure 5 shows a picture of our SunShade prototype with its key components labeled. The algorithm logic runs on a Raspberry Pi, which connects to both an external current sensor—to read changes in current—and to a programmable load to programmatically alter the solar panel's operating voltage. To support model-based WPPT with sensors, our prototype also includes a pyranometer (for sensing solar radiation) and a temperature sensor.

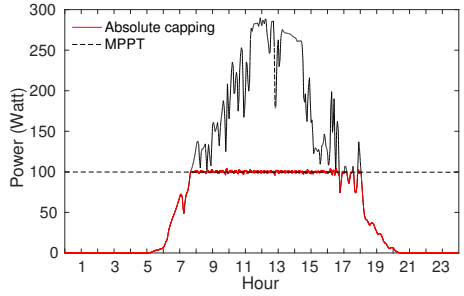


Figure 6: SunShade capping the absolute power of a solar panel to 100W.

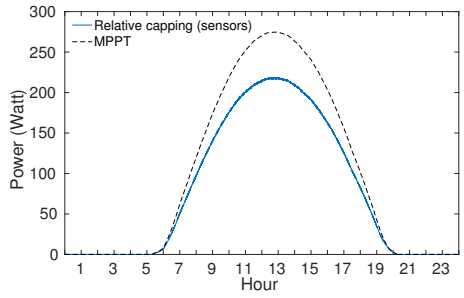


Figure 7: Example of model-based weighted power capping at 80% of the MPP.

5 EVALUATION

We first evaluate the performance and fidelity of SunShade's mechanisms in simulation, as our simulator is able to support a much wider range of experimentation, i.e., covering a range of conditions, compared to our prototype. We then examine the performance and fidelity of our SunShade prototype. To evaluate SunShade's fidelity, we use the Normalized Root Mean Squared Error (NRMSE), which is a common metric for quantifying the difference between two time-series. We compute the NRMSE between SunShade's capped values and the ideal values. In this case, an NMRSE closer to one is better, as it indicates the two time-series are similar. The equation for NMRSE is below, where || denotes the 2-norm of the time-series vector.

$$\text{NRMSE} = 1 - \frac{||actual - estimated||}{||actual - mean(estimated)||} \quad (1)$$

5.1 Simulation Results

Figure 6 demonstrates SunShade's performance in simulation using absolute power capping on a cloudy day. In this case, the clouds are not strong enough to cause the panel output to drop below the cap for most of the middle part of the day. As a result, the system has a near-steady 100W power output with minor oscillations around 100W due to the P&O-based capping algorithm that constantly perturbs voltage searching for the cap. As expected, absolute capping is relatively straightforward as it requires no external knowledge about the MPP. In contrast, WPPT is more challenging, as it requires a prediction (or search) of the MPP.

Figure 7 shows the performance of model-based WPPT (relative capping using sensors) at 80% of the MPP in simulation. Since our model, as with most models, is most accurate when the sun is shining, the 80% cap is a near perfect reflection of 80% of the

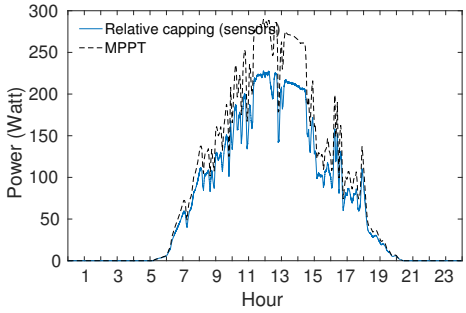


Figure 8: Example of model-based weighted power capping at 80% of the MPP.

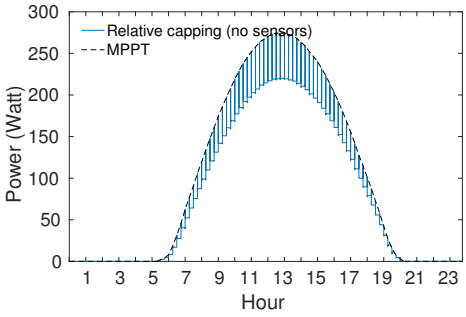


Figure 9: Example of search-based WPPT at 80% MPP on a sunny day.

MPP in real-time. Figure 8 shows the same model-based weighted power capping at 80% on a cloudy day with significant variations in power. The fidelity of WPPT is slightly less (in terms of NRMSE) under cloudy conditions, as the model may be less accurate and there is more time spent searching for the weighted cap. Likewise, Figures 9 and 10 show the performance of the search-based algorithm (relative capping using no sensors) for the same simulated days as above. In this case, we conduct a search at a fixed interval every 15 minutes. The results show that the search-based algorithm frequently deviates from the relative cap to find the MPP in order to reset the cap.

Table 1 shows the NMRSE for each of the mechanisms. The results show that absolute power capping has the highest fidelity, i.e., is closest to the ideal power time-series, since it is the simplest mechanism. In addition, weighted power capping using an accurate model is close to the performance of absolute capping, since it is able to accurately adjust the power cap in real-time without deviating from it. The degradation in fidelity of the approach in our simulations stems primarily from searching for the cap, similar to how MPPT algorithms must search for the MPP. Finally, the last row shows that the NRMSE for search-based WPPT is the lowest for both the sunny and cloudy day. The search-based approach on the sunny day is only slightly lower than the other approaches, despite conducting a search every 15 minutes. However, on the cloudy day the performance is less than the other algorithms, as the 15 minute search interval is too long relative to the frequency of environmental variations.

While the experiments above demonstrate the behavior of our power capping mechanisms for representative sunny and cloudy days, Figure 11 demonstrates how the fidelity of capping changes for

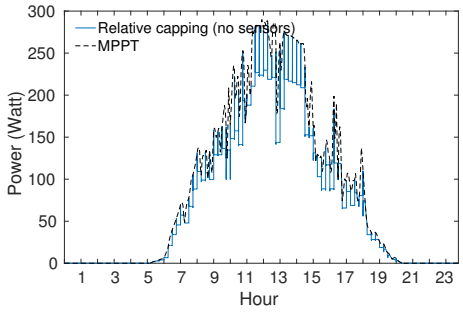


Figure 10: Example of search-based WPPT at 80% MPP on a cloudy day.

Algorithm	Sunny	Cloudy
Absolute	.9888	.9843
WPPT (model-based)	.9874	.9826
WPPT (search-based)	.9677	.8628

Table 1: NRMSE comparison for sunny and cloudy weather conditions

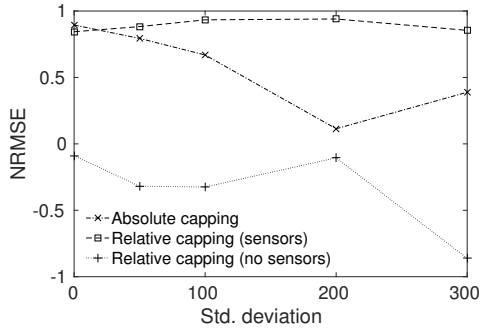


Figure 11: NRMSE for periods with different levels of variation.

the different mechanisms as the frequency of variations increases. In this graph, we subject SunShade to synthetic power fluctuations that increase in frequency along the x-axis. Thus, the higher the frequency on the x-axis the more variance in the power output of the solar panel. The graph shows that for both absolute power capping and model-based WPPT (relative capping using sensors) are much less sensitive to variations than the search-based WPPT (relative capping using no sensors).

5.2 Prototype Results

Due to the complexity of modeling a solar module, we use our SunShade prototype to primarily evaluate the search-based WPPT algorithm. Figure 12 shows power measurements from our SunShade prototype running over a three hour period on a relatively sunny day. The graph includes results from our two systems running in parallel: one running the search-based WPPT (in black) and one running a typical P&O MPPT algorithm (in red). From the MPPT measurements, we compute the ideal WPPT value. In this case, we set the WPPT weight Δ to be 0.5 or 50% of the maximum value. As the graph shows, WPPT periodically searches (every 15 minutes here) for the MPPT to set a new WPPT weight, causing its power output to increase until it converges to the maximum power point, and then to decrease.

At some time periods, clouds cause the WPP to diverge from the ideal between a search interval. For example, near 11am there are passing clouds that cause the MPPT to decrease. During this time period, the WPP also diverges more from the ideal between each search interval. In contrast, after 11:30am, there is little change in the MPP, and thus the WPP tracks the ideal WPP nearly perfectly. This experiment also stressed our WPP implementation due to people passing by the prototype and briefly shading it. This shading is evident in the graph from the periodic dips in both the MPPT and the WPPT. Since these dips caused power output to drop to near zero, they affected both the MPPT and WPPT algorithm even within a search interval. In these cases, WPPT was unable to maintain its power cap due to little available power, causing it to match the MPPT algorithm even without searching for the new MPPT. Note that in a few cases, the passersby shaded the two adjacent panels unevenly, causing only one of them to drop its output.

In addition to the illustrative experiment above, we also examine the impact of changing the search interval, voltage step size, and weight on the fidelity of WPPT compared to the ideal WPPT. Figure 13 shows how the Root Mean Squared Error between WPPT and the ideal WPPT differs as a function of the search interval, which ranges from 30 seconds to three minutes. For each datapoint, we run the WPPT algorithm for 15 minutes with a default voltage step size of 0.5, a default interval of one minute, and a default weight of 50%. Each graph then adjusts one dimension and observes the effect on the RMSE. The proper search interval is a function of the current conditions and the variability of solar output. Under high solar variations, a smaller interval is better, as the ideal weighted cap is changing frequently, while under near constant solar output, a longer interval is better since the MPP is not changing. Here, as the search interval increases, the error also increases, since the weather conditions during this experiment were partly cloudy.

Figure 14 then plots the RMSE as a function of the weight setting. Since lower weights deviate more from the MPP, they take longer to search for the MPP and deviate more from the ideal WPP. The graph demonstrates this trend as the lower the weight setting, the higher the RMSE for our WPPT tracking algorithm. One way to address this issue for lower weights is to save the previous MPP value and immediately start searching from the previous value, rather than from current voltage setting. Of course, this approach is not ideal during highly variable conditions, where the MPP might change significantly. Finally, Figure 15 plots the voltage step size as a function of WPP. Similar to the interval above, the tradeoff in the step size is dependent on the conditions. Under highly variable conditions that require more frequent searching, a larger step size is more desirable as it makes each search faster. Even though the larger step size decreases accuracy, since conditions are highly variable this is outweighed by the faster search time. In contrast, under stable conditions that require fewer searches, a smaller step size is better, since it finds a more accurate weight, which is important because searches occur infrequently. Here, since conditions were variable, we see that, as we increase the voltage step size, the RMSE increases.

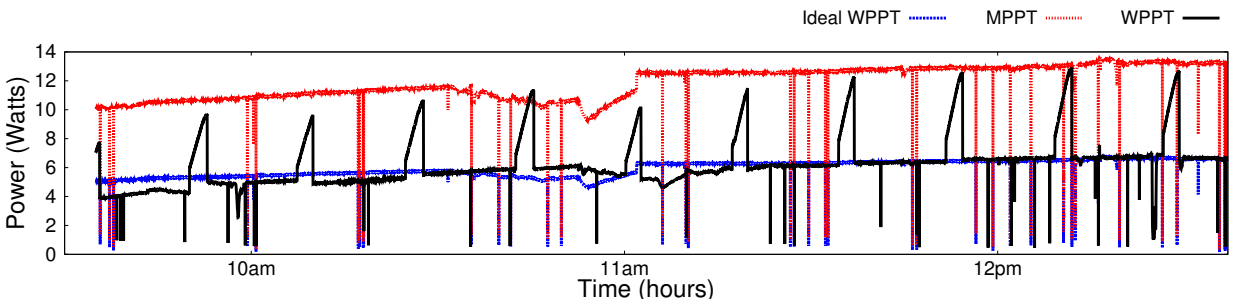


Figure 12: WPPT of SunShade in a real deployment compared with ideal WPPT based on true MPPT

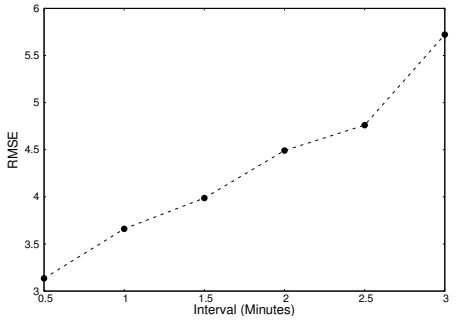


Figure 13: The RMSE between the ideal WPPT and search-based WPPT as a function of the periodic search interval.

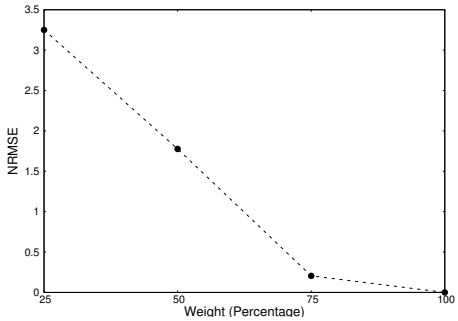


Figure 14: The RMSE between the ideal WPPT and search-based WPPT as a function of the weight.

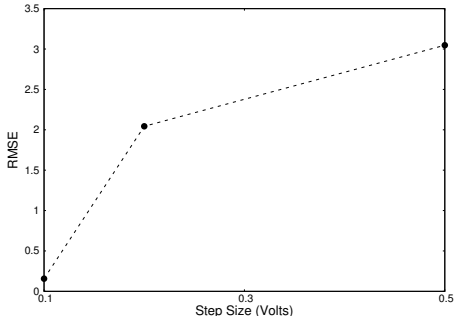


Figure 15: The RMSE between ideal and search-based WPPT as a function of the voltage step-size.

6 RELATED WORK

While prior work has also advocated applying Internet design principles to the overall electric grid [5, 9], SDS systems focus narrowly on applying these design principles to solar systems for multiple reasons: in particular, solar power is the predominant source of distributed generation, is growing rapidly, and is programmatically controllable. Since solar cells are silicon-based semiconductors, their output can be programmatically controlled between zero and their maximum output based on the intensity of light.

SDS and SunShade is related to prior work on active solar power curtailment. However, this work primarily focuses on sensing and responding to specific situations where an inverter may need to reduce or eliminate solar power output. For example, all grid-tied inverters are able to sense a power outage and reduce output to zero to prevent energizing downed power lines. In addition, there is significant prior work on reducing solar power output during over-voltage situations [10, 11, 15–17]. This research differs from our work in that it focuses on specific algorithms and policies embedded in the inverter that respond to specific situations. Instead, our goal is to expose programmatic interfaces to fundamental mechanisms for rate limiting solar power. As a result, SDS decouples mechanism from the policy: while its mechanisms could be used to cap power in response to increased voltage, they could also be used in other contexts.

Recent advancements in smart solar inverters have also recognized the potential benefits of controlling solar power to support grid operation [18]. In contrast, similar to TCP, SDS focuses, not only on managing solar generation to support grid stability, but also on fairly sharing the grid's available capacity to accept solar power with the goal of maintaining grid neutrality. In addition, prior work typically focuses on low-level power systems and power electronics issues. However, the introduction of smart inverters with sophisticated controls is raising the grid's level of abstraction. Similar to the Internet, we expect the grid to evolve into a layered architecture, where the physical layer (layer 2) addresses challenges in power systems and electronics and the higher layers address challenges in capacity management, fair-sharing, quality-of-service, etc. This paper demonstrates mechanisms necessary to address these higher-level problems.

7 CONCLUSION

This paper proposes Software-defined Solar-powered (SDS) systems to control the flow rates of solar power into the grid. The goal of SDS systems is to eliminate the need for policies that artificially cap the number and size of solar deployments that can connect to the grid, and instead dynamically rate-limit them if they exceed capacity in real time. To provide a foundation for SDS systems, this paper presents two fundamental software mechanisms to control solar flow sending rates, including an absolute capping mechanism

and a class of WPPT algorithms that enforce a relative cap. We implement a prototype SDS system, called SunShade, and evaluate the fidelity of these mechanisms for controlling solar flow rates, and their tradeoffs in terms of accuracy and responsiveness.

Acknowledgements. This research is supported by NSF grants IIP-1534080, CNS-1405826, CNS-1253063, CNS-1505422, and the Massachusetts Department of Energy Resources.

REFERENCES

[1] 2014. Solar Energy Industries Association, Solar Energy Facts: 2013 Year In Review. http://www.seia.org. (March 5th 2014).
[2] B. Bendib, H. Belmili, and F. Krim. 2015. A Survey of the Most Used MPPT Methods: Conventional and Advanced Algorithms Applied for Photovoltaic Systems. *Renewable and Sustainable Energy Reviews* 45 (May 2015).
[3] D. Cardwell. 2015. New York Times, Solar Power Battle Puts Hawaii at Forefront of Worldwide Changes. (April 18th 2015).
[4] M. Delfanti, M. Merlo, M. Pozzi, V. Olivieri, and M. Gallanti. 2009. Power Flows in the Italian Distribution Electric System with Dispersed Generation. In *International Conference and Exhibition on Electricity Distribution*. 1–5.
[5] M. He, E. Reutzel, X. Jiang, R. Katz, S. Sanders, D. Culler, and K. Lutz. 2008. An Architecture for Local Energy Generation, Distribution, and Sharing. In *Energy2030*.
[6] IEEE 2014. IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems. https://standards.ieee.org/findstds/standard/1547-2003.html. (2014).
[7] IEEE 2014. IEEE Standard for Interconnecting Distributed Resources with Electric Power Systems, Amendment 1. https://standards.ieee.org/findstds/standard/1547a-2014.html. (2014).
[8] S. Iyengar, N. Sharma, D. Irwin, P. Shenoy, and K. Ramamkritham. 2014. SolarCast - A Cloud-based Black Box Solar Predictor for Smart Homes. In *BuildSys*.
[9] S. Keshav and C. Rosenberg. 2011. How Internet Concepts and Technologies Can Help Green and Smarten the Electric Grid. *CCR* 41, 1 (January 2011).
[10] D. Lew, L. Bird, M. Milligan, B. Speer, X. Wang, E. Carlini, A. Estanqueiro, D. Flynn, E. Gomez-Lazaro, N. Menemenlis, A. Orths, I. Pineda, J. Smith, L. Soder, P. Sorensen, A. Altiparmakis, and Y. Yoh. 2013. *Wind and Solar Curtailment*. Technical Report. National Renewable Energy Laboratory.
[11] R. Tonkoski L. Lopes and T. El-Fouly. 2010. Droop-based Active Power Curtailment for Overvoltage Prevention in Grid Connected PV Inverters. In *IEEE ISIE*.
[12] R. Punyachai, W. Ongsakul, and U. Schmidt. 2014. Impact of High Solar Penetration on Voltage Profiles in Distribution Systems. In *International Conference and Exhibition on Green Energy for Sustainabile Development*. 1–7.
[13] K. Shallenberger. 2016. UtilityDive, Massachusetts Gov. Baker Signs Bill Lifting Solar Cap, Lowering Net Metering Rates. (April 11th 2016).
[14] P. Thomson. 2015. PRI's The World, When the Grid says 'No' to Wind and Solar Power, this Company's Technology helps it say 'Yes' Again. (April 7th 2015).
[15] R. Tonkoski and L. Lopes. 2011. Impact of Active Power Curtailment on Overvoltage Prevention and Energy Production of PV Inverters Connected to Low Voltage Residential Feeders. *Renewable Energy* 36 (June 2011).
[16] R. Tonkoski, L. Lopes, and T. El-Fouly. 2011. Coordinated Active Power Curtailment of Grid Connected PV Inverters for Overvoltage Prevention. *IEEE Transactions on Sustainable Energy* 2, 2 (April 2011).
[17] R. Tonkoski, L. Lopes, and D. Turcotte. 2009. Active Power Curtailment of PV Inverters in Diesel Hybrid Mini-grids. In *IEEE Electrical Power and Energy Conference*.
[18] H. Trabish. 2014. UtilityDIVE, Smart Inverters: The Secret to Integrating Distributed Energy onto the Grid? (June 2014).
[19] J. von Appen, M. Braun, T. Stetz, K. Diwold, and D. Geibel. 2013. Time in the Sun: The Challenge of High PV Penetration in the German Electric Grid. *IEEE Power and Energy Magazine* 11, 2 (March 2013), 55–64.