Bayesian Optimization for Maximum Power Point Tracking in Photovoltaic Power Plants

Hany Abdelrahman, Felix Berkenkamp, Jan Poland, and Andreas Krause

***Abstract*— The amount of power that a photovoltaic (PV) power plant generates depends on the DC voltage that is applied to the PV panels. The relationship between this control input and the generated power is non-convex and has multiple local maxima. Moreover, since the generated power depends on time-varying environmental conditions, such as solar irradiation, the location of the global maximum changes over time. Maximizing the amount of energy that is generated over time is known as the maximum power point tracking (MPPT) problem. Traditional approaches to solve the MPPT problem rely on heuristics and data-based gradient estimates. These methods typically converge to local optima and thus waste energy. Our approach formalizes the MPPT problem as a Bayesian optimization problem. This formalization admits algorithms that can find the maximum power point after only a few evaluations at different input voltages. Specifically, we model the power-voltage curve as a Gaussian process (GP) and use the predictive uncertainty information in this model to choose control inputs that are informative about the location of the maximum. We extend the basic approach by including operational constraints and making it computationally tractable so that the method can be used on real systems. We evaluate our method together with two standard baselines in experiments, which show that our approach outperforms both.**

I. INTRODUCTION

The ability to generate renewable energy from photovoltaic (PV) panels is an important tool towards a solution for problems such as global warming and climate change [1]. An advantage of solar energy is that it can be used in remote locations, where no electricity network is available. To make solar energy more attractive in general, it is important to improve the efficiency of solar panels. Improving the underlying control algorithms is an attractive starting point, as software on existing panels can be updated easily. The challenge for these control algorithms is that the optimal control input varies greatly during the day and depends on the position of the sun, moving clouds, and shadows cast by surrounding buildings [2], which are difficult to model. To achieve maximum power point tracking (MPPT), the control algorithm must track the global optimum of a changing, non-convex function with changing optima. State-of-the-art algorithms used in the industry rely on gradient ascent methods, which typically only converge to local maxima and thereby waste energy. In contrast, the MPPT algorithm that we introduce in this paper is able to track the maximum power point (MPP) globally and thereby generates significantly more energy.

At the heart of MPPT lies the problem of finding the optimum of an unknown function, the power-voltage curve. In general, any existing optimization technique can be used. The amount of power generated depends, to large parts, on the properties of the optimization method. Many different methods have been evaluated in the literature [3], [4]. The most common techniques used are gradient ascent algorithms, which operate by perturbing the applied voltage periodically until a local power optimum is found. To avoid oscillations around the optimum, variable steps sizes have been explored in [5] and [6]. However, the main problem with gradient-based methods is that they waste energy, since they typically converge only to a local optimum.

Other methods that have been explored for MPPT include fuzzy logic control in [7] and [8]. However, the resulting performance depends on approximations and the manual construction of a rule base. Another method is current sweep [9], which computes the power-voltage curve directly by sweeping through all input voltages. This method is infeasible for large plants, wastes significant amounts of energy during the sweep, and cannot track fast changes in the MPP caused by changing environmental conditions.

More recently, learning approaches have been considered. In particular, neural networks have been used to learn optimal voltage inputs depending on measurements of irradiation, temperature and various parameters of the PV array [3]. The main disadvantage of neural networks is that they require large amounts of training data and are specific to particular PV models and atmospheric conditions. If a certain condition is encountered that has not been part of the training set, the neural network does not pick optimal actions.

A method, which has not been considered for MPPT previously, but which is popular in the area of machine learning, is Bayesian optimization. In general, Bayesian optimization has the goal of finding the global optimum of an unknown function within few evaluations. One popular approach is to model the unknown function as a Gaussian process (GP) [10]. GP models provide a mean estimate and associated uncertainty information, which is used in Bayesian optimization to guide future function evaluations to locations that are informative about the maximum [11]. For example, this method has been used to optimize linear quadratic controllers on a real system [12] and to optimize controller parameters subject to safety constraints [13], [14].

Our approach for MPPT is based on Bayesian optimization and models the power-voltage curve as a GP. The

Felix Berkenkamp, and Andreas Krause are with the Learning & Adaptive Systems Group (LAS), Department of Computer Science, ETH Zurich, Switzerland. Email: {befelix, krausea}@ethz.ch

Hany Abdelrahman is with the Department of Computer Science, ETH Zurich, Switzerland. Email: hanya@ethz.ch

Jan Poland is with ABB Corporate Research, Switzerland. Email: jan.poland@ch.abb.com

voltage inputs are selected with the GP-*Upper Confidence Bound* (GP-UCB) algorithm [11]. We extend this algorithm to comply with operational and computational constraints. The result is an efficient and feasible algorithm that can exploit previous knowledge to find and track the global MPP. In contrast to the previously mentioned approaches, our MPPT algorithm does not rely on a manually designed rule base, deals gracefully with new and changing environmental conditions and generates more energy.

II. MODEL

In this section, we review the model of a single PV power cell. We use a series combination of these models in order to evaluate our method experimentally in Sec. V.

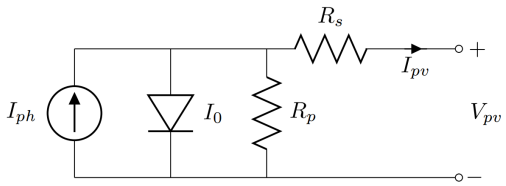

Fig. 1. Equivalent circuit of a PV cell. The control input is the voltage V_{pv}, while the current I_{ph} is dictated by environmental conditions.

An illustration of the model of a single photovoltaic cell according to [2] is shown in Fig. 1. Environmental conditions determine the generated current I_{ph}, which is modeled as a function of solar irradiance G, and ambient temperature T,

$$I_{ph}(G,T) = \big(I_{sc} + K_i(T - T_r)\big)\frac{G}{G_r}, \tag{1}$$

where the parameters specific to the PV cell are the short circuit current I_{sc}, the temperature coefficient K_i, the reference irradiance G_r, and the reference temperature T_r.

We model the current I_0 that flows through the diode as

$$I_0 = I_{01}\left(\exp\left(\frac{q(V_{pv} + I_{pv}R_s)}{Ak_bT}\right) - 1\right), \tag{2}$$

which depends on the diode saturation current I_{01}, the diode ideality factor A, the electron charge constant q, and the Boltzmann constant k_b.

By applying Kirchoff's law on the top-left node in Fig. 1, the PV current I_{pv} is given by

$$I_{pv} = I_{ph} - I_0 - \frac{V_{pv} + I_{pv}R_s}{R_p}. \tag{3}$$

The current I_{pv} of a single photovoltaic cell can be obtained by solving (3) and the generated power is equal to $P_{pv} = V_{pv}I_{pv}$.

A. Power Plants and Partial Shading

In large PV power plants, usually several PV cells are connected in series. For safety reasons, the individual PV cells are connected in parallel to a separate bypass diode. In typical conditions, not all PV cells in such a series connection receive the same amount of irradiance. This is known as partial shading and can be caused, for example, by clouds or shadows cast by the surrounding environment. During partial shading, highly illuminated cells generate more current than less illuminated ones. Since the current flow between all PV cells in a series connection has to be equal, excess current, which is generated by highly illuminated PV cells, flows through the bypass diode.

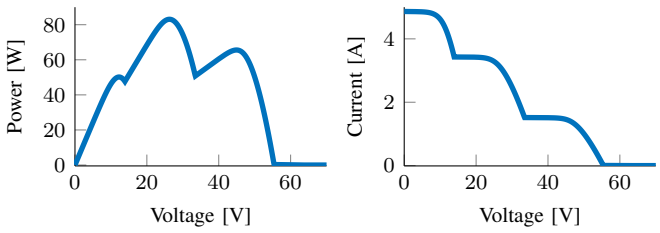

Fig. 2. Examples of power-voltage and current-voltage curves under partial shading. The power-voltage curve has multiple local maxima, which are challenging for gradient ascent methods.

The only directly controlled variable is the voltage that is applied to the entire series connection. This voltage determines the individual voltages and currents of the PV cells through a nonlinear interaction based on (3). As a consequence of this interaction, the power-voltage curve has multiple local maxima during partial shading, which can be seen in Fig. 2.

III. PROBLEM STATEMENT

The goal of MPPT is to maximize the power generated by an array of PV cells that are connected in series. Equivalently, we aim to minimize the power lost by not operating the plant at the time-varying, optimal voltage v_t^* that maximizes the generated power. The optimal input voltage v_t^* is not known *a priori*. Moreover, it is not possible to use the model in Sec. II in order to compute v_t^* without a significant system identification effort and access to sensors at every module to infer all the relevant environmental variables. We formalize the control problem as a regret minimization problem,

$$\min_{v_t \in \mathcal{A},\, t \geq 1} \sum_{t=1} p_t(v_t^*) - p_t(v_t), \tag{4}$$

which corresponds to minimizing the amount of power lost by, at time t, operating the plant at voltage v_t that is different from the optimal voltage v_t^*. The power-voltage curve at time t is given by $p_t(\cdot)$ and input voltages can be selected from the set $\mathcal{A}$. The formulation in (4) is equivalent to maximizing the 1-norm of the power signal.

IV. BACKGROUND

In this section, we introduce the Bayesian optimization algorithm that we use to optimize the amount of energy generated. As a first step, we introduce Gaussian processes (GPs) as a model for the power-voltage curve, $p_t(\cdot)$ in (4). We use this model in our MPPT algorithm in order to estimate the optimal control voltage, v_t^*.

A. Gaussian Processes (GPs)

GPs are a popular choice for nonparametric regression in machine learning, where the goal is to find an approximation of a nonlinear map $p(v)\colon \mathcal{A} \to \mathbb{R}$ from an input vector $v \in \mathcal{A}$

to the function value $p(v)$. The main assumption is that function values $p(\cdot)$ associated with different voltage inputs are random variables and have a joint Gaussian distribution. This distribution is specified by a mean function, which is assumed to be zero without loss of generality, and a covariance function, $k(v, v')$. The covariance function is also known as the kernel and can be interpreted as a measure of similarity between any two voltage values, v and v'. It encodes assumptions about the underlying function. For common kernel choices and their interpretations see [10]. We provide more information about the kernel used in this work in Sec. V.

The GP framework can be used to predict the generated power, $p(v)$, at an arbitrary input voltage, $v \in \mathcal{A}$, based on a set of t past observations, $y_t = \left[\hat{p}(v_1), \ldots, \hat{p}(v_t)\right]^{\mathrm{T}}$ at voltages $\mathcal{A}_t = \{v_1, \ldots, v_t\}$. The observations of the function values, $\hat{p}(v_t) = p(v_t) + \omega_t$, are corrupted by Gaussian noise, $\omega_t \sim \mathcal{N}(0, \sigma^2)$. Conditioned on these observations, the mean and variance of the prediction at v are given by

$$\mu_t(v) = k_t(v)(K_t + \mathbb{I}_t\sigma^2)^{-1}y_t, \tag{5}$$

$$\sigma_t^2(v) = k(v, v) - k_t(v)(K_t + \mathbb{I}_t\sigma^2)^{-1}k_t^{\mathrm{T}}(v), \tag{6}$$

where the vector $k_t(v) = \left[k(v, v_1), \ldots, k(v, v_t)\right]$ contains the covariances between the new input, v, and the past data points in $\mathcal{A}_t$, the covariance matrix, $K_t \in \mathbb{R}^{t\times t}$, has entries $[K_t]_{(i,j)} = k(v_i, v_j)$ for $i, j \in \{1, \ldots, t\}$, and the identity matrix is denoted by $\mathbb{I}_t \in \mathbb{R}^{t\times t}$.

B. Optimization Algorithm

In this section, we show how the GP model of the power-voltage curve from the previous section can be used to find the optimal operating voltage and maximize the generated power for the static case; that is, the power-voltage curve does not change over time. We extend this to the dynamic case in Sec. IV-C.

In general, any Bayesian optimization algorithm is designed to find the global optimum of an unknown function within few evaluations on the real system. The algorithm that we use here is the *GP-Upper Confidence Bound* (GP-UCB) algorithm [11], which formalizes the problem as a multi-armed bandit problem [15]. This means that at each time step t we pick one voltage and observe a corresponding, noisy measurement of the generated power.

The goal of the GP-UCB algorithm is to minimize the lost power in (4). Since the optimal voltage input, v_t^*, is not known *a priori*, the optimal strategy has to balance learning about the location of the MPP (exploration), and selecting a voltage that is known to lead to high power generation (exploitation). GP-UCB uses the GP prediction and associated uncertainty in (5) and (6) to trade off between exploration and exploitation by selecting the voltage at time step t according to

$$v_t = \underset{v\in\mathcal{A}}{\operatorname{argmax}}\ \mu_{t-1}(v) + \beta_t^{1/2}\sigma_{t-1}(v), \tag{7}$$

where β_t is a scalar that determines the confidence interval. The intuition behind this algorithm is that it selects the voltage for which the upper confidence bound of the GP model

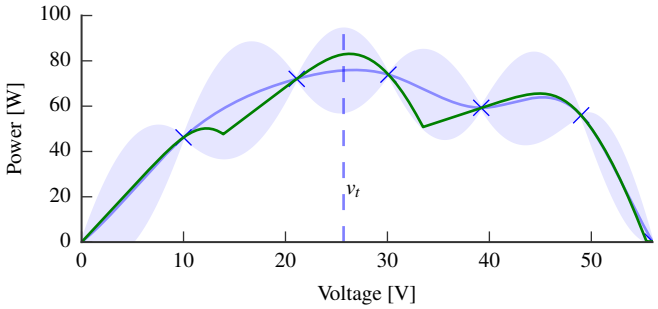

Fig. 3. Example plot for GP-UCB. At iteration t, the algorithm chooses to apply the input voltage v_t based on the GP posterior over the power-voltage curve (mean and confidence interval in blue, unknown, true function in green). Iteratively selecting voltages with the maximum upper-confidence bound decreases the GP's uncertainty about the true function at positions that could be the maximum until the global maximum is found.

of the generated power is maximal. Repeatedly evaluating the system at voltages given by (7) improves the mean estimate of the underlying function and decreases the uncertainty at candidate locations for the maximum, such that the global maximum is found eventually [11]. After each time step, we obtain a new measurement of the generated energy and update the GP model. An example of how (7) chooses the input voltage can be seen in Fig. 3.

The GP-UCB algorithm guarantees that we find the global optimum of the power-voltage curve after few evaluations and thereby maximize the generated power. Moreover, the algorithm has strong theoretical bounds on the amount of energy that is lost in (4), which is known as cumulative regret in the machine learning literature [11]. These theoretical guarantees make the GP-UCB algorithm a favorable choice for MPPT and other extremum seeking control applications.

C. Contextual Optimization Algorithm

The GP-UCB algorithm assumes that the underlying function does not change during the optimization procedure. However, the power-voltage curve, $p_t(v)$, and consequently the MPP, v_t^*, change with environmental conditions. The *Contextual* GP-UCB (CGP-UCB) algorithm considers this scenario [16]. It is a conceptually simple extension of the GP-UCB algorithm. Instead of defining the kernel of the GP in Sec. IV-A only in terms of the input voltage, the function is also allowed to vary depending on a context. For example, this context can be the time or a measurement of the irradiance. We give more details on the context used in our experiments in Sec. V-B.

D. Streaming Algorithms

The CGP-UCB algorithm discussed in the previous section is able to find the global optimum of the changing power-voltage curve. However, the computational complexity increases over time, $O(t^3)$, as larger kernel matrices have to be inverted in order to compute (5) and (6). In order to overcome this problem, a simple heuristic is to only use a certain time window of the last data points. The disadvantage of this approach is that the algorithm cannot learn from data beyond this time window in order to perform better

under recurring environmental conditions. An alternative is to summarize past data points on the fly with a smaller set of points, $\mathcal{S}$. One way to do so, similar to the approach in [17], is to maximize a measure of information,

$$f(S) = \frac{1}{2}\log\det(\mathrm{I}_S + \lambda^2 K_{\mathcal{S}}), \tag{8}$$

where $K_{\mathcal{S}}$ is the kernel matrix of the summarized data in S, and λ is a regularization parameter. It is possible to maximize (8) efficiently and with tight approximation guarantees by using the *Stream Greedy* algorithm. For more details on the algorithm and the theoretical properties see [18]. We evaluate this approach in our experiments in Sec. VI-B.

V. EXPERIMENTAL SETUP

In this section, we introduce the experimental setup that we use to to evaluate our approach in Sec. VI.

A. Simulation with Real Data

In order to evaluate our method in a realistic scenario, we simulate the model in Sec. II based on recordings of solar irradiance sensors from the department of electrical engineering at Tampere University of Technology. The data is from June 4th, 2014, a cloudy day with maximum solar irradiance of $1172\,\mathrm{W/m^2}$. We use the power plant model from Sec. II to simulate a plant with 6 strings of 10 PV cells connected in series with a rated power of 3 kWp. The control input is computed at a frequency of 5 Hz.

Due to the small dimensions of the research facility, partial shading has significantly less impact than in large scale plants. In order to produce results based on the available data that are more relevant to large scale plants and thus more meaningful in practice, the simulated area of the plant has been artificially enlarged by applying a constant scaling to the differences of the sensors at the end of each string.

In order to have a fair comparison, we split the data into a training and test set. The training set is used to determine the parameters of the GP model in Sec. IV-A, while the test set is used to compare our approach to other baselines.

B. Kernel Selection for the Gaussian Process Model

In order to use the CGP-UCB algorithm from Sec. IV-C, we have to choose an appropriate kernel for the GP that models the power-voltage curve and how it changes depending on the context (the environment).

We choose a multiplication of two kernels, $k = k_z \otimes k_v$, where $k_v \colon \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ is a kernel that models how the generated power changes depending on the input voltage and $k_z \colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{R}$ is a kernel for the context, $z \in \mathcal{Z}$. This context can be anything general, more details on the specific choice are given below. The intuition behind the multiplicative kernel choice is that one would expect the generated power to be similar only if both the input voltages and the environmental conditions are similar.

In general, there is a lot of flexibility in the choice of kernel and it can be used to encode specific prior knowledge about the kind of functions one may expect to occur. In this work, we choose a Matérn kernel to model effects of changing voltages,

$$k_{\mathrm{mat}}(v, v') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2v}(v-v')}{l}\right)^{\nu} K_\nu\left(\frac{\sqrt{2v}(v-v')}{l}\right), \tag{9}$$

where $\Gamma(v)$ is the gamma function, K_v is the modified Bessel function of the second kind, and l is a scaling parameter. The parameter ν encodes how often the underlying function is expected to be differentiable. Here we chose $\nu = 2.5$ in order to model sudden changes in the power when the voltage changes, as can be seen in Fig. 2. Additionally, we encode prior knowledge about known shapes of power-voltage curves by using two different, explicit basis functions, $\phi_{1/2}(v)$, scaled by a learned factor α. These basis functions were computed using Principal Component Analysis (PCA) on the training data set and allow us to encode previous knowledge about commonly occurring function shapes. The resulting combined kernel for the voltage is given by

$$K_v(v, v') = \alpha \sum_{i=1}^{2} \phi_i(v)^T \phi_i(v') \; + K_{\mathrm{mat}}(v, v'). \tag{10}$$

To model changes in the environment, we have to select a context z that reflects important factors that impact the generated power. In this work we compare two different choices. Firstly, we can use time as a context. The advantage is that this does not require additional sensors or knowledge about which factors impact the power-voltage curves. However, this model cannot recognize similar environmental conditions and, as a result, does not benefit from past experience beyond a certain time horizon. As a second approach, we assume that the solar irradiance is the main factor that influences the voltage curve and that sensor readings are available. We use six sensors that a spread out over the plant as a context.

For both contexts we use a squared exponential kernel,

$$k_{\exp}(z, z') = \sigma^2 \exp\left(\frac{-(z-z')^2}{2\bar{l}^2}\right), \tag{11}$$

which represents our assumption that environmental effects will change the power-voltage curve more smoothly than changes in the voltage. Here, $\bar{l}$ is a scaling parameter.

For both kernels, the hyperparameters, e.g., the scaling parameters, were learned from the training data set using a mixture of cross-validation and maximum marginal likelihood methods. The β_t parameter in (7) was set according to Theorem 1 in [11], but scaled down by a factor of 5 to increase empirical performance. Similar results can be obtained with constant values of β_t.

C. Comparison Baselines

In this section, we introduce two baselines that are commonly used in the industry. The first baseline is the *Perturb and Observe* (P&O) algorithm [3]. It is a simple hill-climbing method that follows the direction that increases power relative to the last observation. The step size for this algorithm

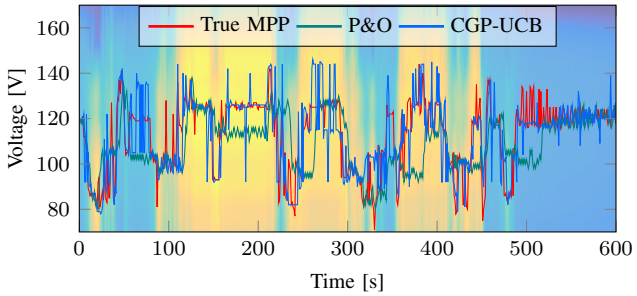

Fig. 4. The true MPP (red) is tracked by the CGP-UCB algorithm (blue) over a 10 minutes subset of the data. The CGP-UCB algorithm is able to track the globally optimal MPP reliably. Spikes away from the true MPP occur whenever the algorithm needs to gather additional information to determine the location of the MPP.

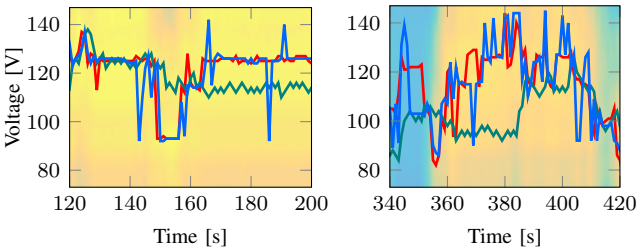

Fig. 5. Zoomed-in sections of Fig. 4. The P&O baseline gets stuck in local optima and thereby wastes energy. The CGP-UCB algorithm can escape local optima reliably.

was learned from the training data set. This method, although simple, is still an area of active research [19].

The second baseline is the *Incremental Conductance* (IC) algorithm [3]. It is based on the observation that the slope of the power voltage curve is zero at the MPP, positive to its left, and negative to its right. The slope is given by

$$\frac{\partial p(v)}{\partial v} = \frac{\partial (iv)}{\partial v} = i + v\frac{\partial i}{\partial v} \approx i + v\frac{\Delta i}{\Delta v}, \qquad (12)$$

where i is the current flowing through the PV module. The MPP can be tracked by comparing the instantaneous conductance, i/v, to the incremental conductance, $\Delta i/\Delta v$. The MPP is determined to be reached when $\partial p/\partial v = 0$.

VI. EXPERIMENTAL RESULTS

In this section, we compare our method to the two baselines that we introduced in Sec. V-C. Furthermore, we show how to incorporate computational and operational constraints.

An illustrative 10-minute experiment of CGP-UCB based on data of the test set can be seen in Fig. 4. The background image is a heat map of the power-voltage curves, where the power-voltage curve at each time step is indicated by the background color. Blue regions represent time instants with significant shading and low output power, while in the yellow regions output power is relatively high. It can be seen that the CGP-UCB algorithm is able to track the MPP reliably. Occasionally, the algorithm evaluates voltages away from the true, unknown MPP, which causes spikes in the blue line. This behavior occurs whenever the GP model of the power-voltage curve does not have sufficient knowledge

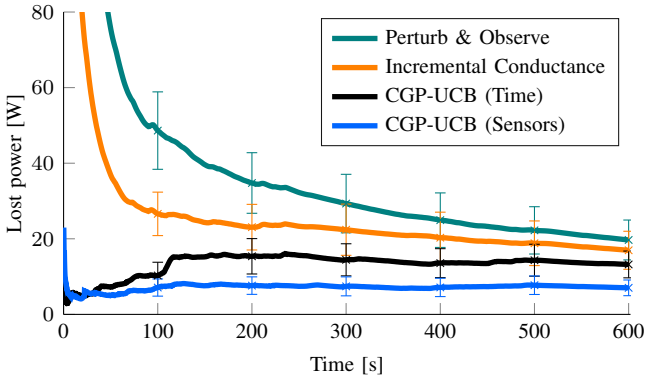

Fig. 6. Average lost power (regret) and standard error of 10 different experiments for the two baselines and the CGP-UCB algorithm based on time and sensor measurements. The CGP-UCB algorithm outperforms both baselines. The version based on sensors has access to additional information relative to the other three methods and wastes less energy.

about the underlying function in order to be confident about the location of the MPP. In these cases the algorithm explores other voltages until sufficient knowledge about the MPP has been regained.

In Fig. 5 we show two zoomed-in sections of Fig. 4 along with the P&O baseline. Since the baseline is based on gradient-ascent, it gets stuck in a local optimum of the power-voltage curve and thereby does not generate the maximum power possible. Moreover, we can see that the P&O baseline keeps oscillating around the local optimum that it has found. In comparison, the GP-UCB algorithm can escape local optima and track the MPP.

Since the two baselines are not able to track the global optimum, they waste energy compared to the CGP-UCB algorithm based on either time or irradiance measurements as a context. The average power wasted by not operating the plant at the true MPP and associated standard error over 10 different experiments can be seen in Fig. 6, where the lost energy is the area underneath each curve. It can be seen that the baselines waste significant amounts of energy by converging to local optima. Both, the CGP-UCB algorithm based on time and sensor measurements as a context outperform the baselines. The CGP-UCB based on sensors has access to additional information and can exploit previously observed environmental conditions. As a consequence, it outperforms the other algorithms that are based only on time. The large difference between the algorithms at the beginning of the experiments occurs, because the baselines take time to converge to a local MPP from the initial voltage. There is no good guideline to select the starting point for the two baselines. The values shown in Fig. 6 are the average over 100, linearly-spaced initial voltages.

A. Operational Constraints

So far, we have considered a plant model without constraints. In practice, PV power plants have operational constraints, such as limitations on how fast the voltage can be changed from one value to another. This is due to the presence of large capacitors in the system, which take time to charge or discharge. The original CGP-UCB algorithm in

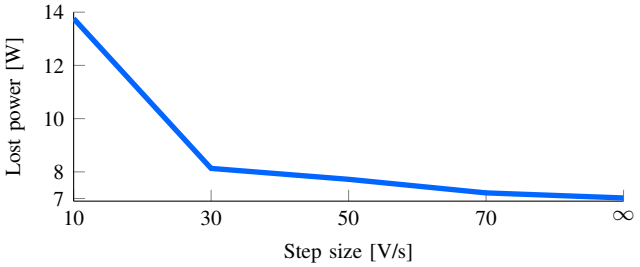

Fig. 7. Average wasted power, as in Fig. 6, after including voltage constraints for GP-UCB based on sensors. Including operational constraints does not degrade performance for feasible step sizes, i.e. 50-70 V/s.

Sec. IV-C assumes no constraints. Here, we limit the allowed voltage change, which corresponds to a limit on the step size between time steps.

We introduce a simple heuristic that has proven to work well in practice. We determine the next voltage to apply at each time step according to (7), as in the normal CGP-UCB algorithm. However, instead of directly jumping to this voltage, we take a step towards it, which is limited by the maximum step size. The impact of this step size on the average wasted energy during the same experiments as in Fig. 6 is shown in Fig. 7. It can be seen that, for large step sizes, the algorithm performs similarly to the unconstrained case, as one would expect. However, even for step sizes that are realistic in typical plants, such as 50-70 V/s, the amount of additionally wasted energy is small.

B. Computational Efficiency

Next to the operational constraints, we consider the impact of using the streaming algorithm from Sec. IV-D on sensor measurements. In this case, rather than using the entire data history, we use a fixed number of data points. The goal is for this set to be informative about the current location of the MPP and, additionally, to summarize past data effectively in order to provide information about reoccurring conditions. To this end, we use the most recent measurements for a part of these data points, while the rest is selected to summarize the past data, see Sec. IV-D. We compared several different set sizes and combinations of past and recent data. The best performance was achieved with a set size of 200 data points, where 100 data points were used both for the most recent points and to summarize the past. This method performed within the standard error of the approach that is able to access all the past data. At the same time, this method makes the algorithm computationally viable in practice.

VII. CONCLUSIONS

In this work, we presented the maximum power point tracking problem as a Bayesian optimization problem. We used a state-of-the-art Bayesian optimization algorithm, CGP-UCB, in order to select input voltages that maximize the amount of generated energy. We extended the algorithm to be applicable to real-world applications and compared it to two baselines that are commonly used in the industry. Experiments based on measurement data in a realistic setting showed that our approach generates significantly more energy than those baselines in situations where parts of the power plant are shaded.

ACKNOWLEDGEMENT

We would like to thank Kari Lappalainen from Tampere University of Technology for providing the solar irradiance dataset that has been of great help in this research. This research was supported by SNSF grant 200020_159557.

REFERENCES

[1] Z. Şen, "Solar energy in progress and future research trends," *Progress in energy and combustion science*, vol. 30, no. 4, pp. 367–416, 2004.
[2] M. Seyedmahmoudian, S. Mekhilef, R. Rahmani, R. Yusof, and E. T. Renani, "Analytical modeling of partially shaded photovoltaic systems," *Energies*, vol. 6, no. 1, pp. 128–144, 2013.
[3] T. Esram, P. L. Chapman *et al.*, "Comparison of photovoltaic array maximum power point tracking techniques," *IEEE Transactions on Energy Conversion*, pp. 439–449, 2007.
[4] S. Ovaska, "Maximum power point tracking algorithms for photovoltaic applications," Ph.D. dissertation, Aalto University, 2010.
[5] W. Xiao and W. G. Dunford, "A modified adaptive hill climbing MPPT method for photovoltaic power systems," in *Proc. of the Power Electronics Specialists Conference*, vol. 3, 2004, pp. 1957–1963.
[6] A. Al-Amoudi and L. Zhang, "Optimal control of a grid-connected PV system for maximum power point tracking and unity power factor," in *Proc. of the International Conference on Power Electronics and Variable Speed Drives*, 1998, pp. 80–85.
[7] N. Khaehintung, K. Pramotung, B. Tuvirat, and P. Sirisuk, "Risc-microcontroller built-in fuzzy logic controller of maximum power point tracking for solar-powered light-flasher applications," in *Proc. of the IEEE Industry Electronics Society Conference*, vol. 3, 2004, pp. 2673–2678.
[8] R. M. Hilloowala and A. M. Sharaf, "A rule-based fuzzy logic controller for a pwm inverter in photo-voltaic energy conversion scheme," in *Proc. of the IEEE Industry Applications Society Meeting*, 1992, pp. 762–769.
[9] M. Bodur and M. Ermiş, "Maximum power point tracking for low power photovoltaic solar panels," in *Proc. of the IEEE Mediterranean Electrotechnical Conference*, 1994, pp. 758–761.
[10] C. E. Rasmussen, "Gaussian processes for machine learning." MIT Press, 2006.
[11] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: no regret and experimental design," in *Proc. International Conference on Machine Learning (ICML)*, 2010, pp. 1015–1022.
[12] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, "Automatic LQR tuning based on Gaussian process global optimization," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
[13] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016, arXiv:1509.01066 [cs.RO].
[14] F. Berkenkamp, A. Krause, and Angela P. Schoellig, "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics." arXiv, 2016, arXiv:1602.04450 [cs.RO].
[15] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, vol. 47, no. 2-3, pp. 235–256, 2002.
[16] A. Krause and C. S. Ong, "Contextual Gaussian process bandit optimization," in *In Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2011, pp. 2447–2455.
[17] M. Seeger, C. Williams, and N. Lawrence, "Fast forward selection to speed up sparse Gaussian process regression," in *Proc. of the International Workshop on Artificial Intelligence and Statistics*, 2003.
[18] R. Gomes and A. Krause, "Budgeted nonparametric learning from data streams," in *Proc. International Conference on Machine Learning (ICML)*, 2010, pp. 391–398.
[19] L. Piegari and R. Rizzo, "Adaptive perturb and observe algorithm for photovoltaic maximum power point tracking," *IET Renewable Power Generation*, vol. 4, no. 4, pp. 317–328, 2010.