Chapter 5

A Quick Maximum Power Point Tracking Method Using an Embedded Learning Algorithm for Photovoltaics on Roads

Koichiro Yamauchi

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.79711

Abstract

This chapter presents a new approach to realize quick maximum power point tracking (MPPT) for photovoltaics (PVs) bedded on roads. The MPPT device for the road photovoltaics needs to support quick response to the shadow flickers caused by moving objects. Our proposed MPPT device is a microconverter connected to a short PV string. For real-world usage, several sets of PV string connected to the proposed microconverter will be connected in parallel. Each converter uses an embedded learning algorithm inspired by the insect brain to learn the MPPs of a single PV string. Therefore, the MPPT device tracks MPP via the perturbation and observation method in normal circumstances and the learning machine learns the relationships between the acquired MPP and the temperature and magnitude of the Sun irradiation. Consequently, if the magnitude of the Sun beam incident on the PV panel changes quickly, the learning machine yields the predicted MPP to control a chopper circuit. The simulation results suggested that the proposed MPPT method can realize quick MPPT.

Keywords: photovoltaics bedded on road, embedded learning algorithm, incremental learning, insect brain, modal regression on a fixed memory budget, maximum power point tracking (MPPT), shadow flicker, partial shading, micro converter

1. Introduction

In recent years, renewal energy technologies have attracted considerable attention as they prevent degradation of the environment to a large extent. Photovoltaics (PVs) are one such technology. However, the drawbacks of photovoltaic systems are that they are unstable while generating electricity and that they require a wide area to catch a large amount of sunlight.

One solution is to place photovoltaics on roads. As the total area covered by roadways in the world is extremely high, it is worth using it as PV sites. Still, objects moving on the road cause shadows. In particular, the shadow flickers on PV systems cause power conditioners connected to the PVs to behave in an unstable manner. Such unstable behavior forms the origin of degradation and greatly reduces the amount of electricity generated.

As shown in Section 2, PVs demonstrate highly nonlinear characteristics and its maximum power point cannot be analytically derived. Therefore, maximum power point tracking (MPPT) devices track MPP using various heuristics. As mentioned in previous survey papers [1, 2], the most preliminary technique for realizing MPPT is the perturbation and observation (P&O) method. P&O is a type of hill-climbing algorithm. The P&O method provides a perturbation to the current and the voltage and checks whether the output power increases. If the power has increased, the P&O method employs the same voltage change in the next step and vice versa. Although the P&O method is easy to implement within small embedded systems, there is no guarantee that the perturbed voltage is suitable for obtaining MPP. The incremental conductance (IncCond) [3] and the ripple correlation (RCC) methods [4] overcome this problem by estimating the gradient of the power curve. These two methods can be realized in analog circuits and can demonstrate quick convergence behaviors. Fuzzy logic control methods are also usually used for controlling the change in duty ratio for the chopper circuit. Fuzzy logic controllers can work appropriately even if its inputs are ambiguous, and they show a quick convergence behavior to the MPP. For example, a previous paper [5] demonstrated the use of fuzzy logic that yielded a change in the duty ratio from the difference between the current photovoltaic output voltage and the predicted MPP. Neural network-based MPPT methods are also proposed (e.g., [6]). The model predicts MPP and its corresponding maximum current using a pretrained neural network. The model cannot adjust its neural network for changing environments. In our previous study, a hybrid system involving the P&O method and an embedded learning machine was constructed [7]. The learning machine studies the MPP acquired by the P&O method when solar irradiation is stable. When solar irradiation changes quickly, the learning machine predicts MPP. However, these methods do not support MPPT under an inhomogeneous isolation condition, where the voltage-power curve has several local peaks.

Recently, a particle swarm optimization (PSO)-based MPPT method was proposed [8]. This method can estimate all local power peak points and select the best one. However, the resultant solutions are highly depending on the initial particles.

On the contrary, a previous study [9] demonstrated that a swing technique can acquire the voltage-power curve by scanning within a certain short interval. It shorts, the series-connected PV string and an inductor simultaneously observe the voltage and power until the output voltage reaches zero. Therefore, the device can detect MPP during the scan. However, it needs special hardware to realize the swing.

To overcome this problem, we use a quick converter connected to a PV string. The main challenge here is finding the MPP from the complex power-voltage curve.

In our previous study [7], we proposed a model that uses an incremental learning method based on general regression neural network. The method is used to obtain the magnitude of solar irradiation s_t, temperature T_t, and MPP derived by the P&O method. Although the system

http://dx.doi.org/10.5772/intechopen.79711

quickly detects MPPs of a single solar panel, which has a single cluster, it cannot detect the MPP of solar panels with several clusters or solar panels connected in series.

In this chapter, we propose an MPPT converter that detects MPPs of solar panels with several clusters using a modal regression method on a fixed memory budget. To realize quick MPPT, the proposed method uses a learning machine on a fixed memory budget. The learning machine on a fixed budget is a small learning machine that can continue online learning on a fixed storage space. Therefore, it is suitable to be embedded to a small microcomputer. The learning on a budget should be executed on a system with a small amount of storage space with low computational power.

To this end, it is worth referencing the mechanisms of an insect's brain. Although the precise mechanism of an insect's small brain that is a source of their intelligence is not known, it is true that their sensory system is much smaller than that of humans. Therefore, the dimensions of their sensory inputs are small. As mentioned in Section 3.2, the storage space for recording the kernels is proportional to the number of input dimensions. From this insight, we should be able to reduce the input dimensions to reduce the storage space for the learning machine.

The rest of the chapter is organized as follows. Section 2 describes the photovoltaic properties, and Section 3 introduces an MPPT algorithm accelerated by a learning machine using a modal regression on a budget. Section 4 shows computer simulation results of the new MPPT algorithm, and Section 5 concludes this chapter.

2. Properties of photovoltaics

Photovoltaics are a type of current sources, whose current flow is determined by the strength of solar irradiation. A normal solar panel comprises several photovoltaic cells. These cells are usually connected in series, and the series-connected cells are then connected in parallel. Such solar panels show highly nonlinear characteristics and is usually modeled by using the following equation [10, 11]. Let us denote the output voltage and current from the photovoltaic as V_{pv} and I_{pv}, respectively. According to the equivalent circuit shown in **Figure 1**, I_{pv} is represented by (1).

$$\mathrm{I_{pv}} = \mathrm{N_p} I_{sc}\left(\frac{I_r}{100}\right) - \mathrm{N_p} I_o\left[\exp\left(\frac{qV_{pv}}{nkTN_s}\right) - 1\right], \tag{1}$$

where V_{pv}, the terminal voltage of the photovoltaic [V]; I_{pv}, output current from the photovoltaic [A]; I_p, photocurrent [A]; I_o, saturation current [A]; I_{sc}, short-circuit current [A]; I_r, irradiation [%]; n, ideality factor; q, charge of electron [C]; k, Boltzmann's constant; T, junction temperature [$^\circ$C]; $\mathrm{N_p}$, number of cells in parallel; $\mathrm{N_s}$, number of cells in series.

In Eq. (1), I_r is given by the ratio of actual strength of solar irradiation to the irradiation of standard test condition [11]. Therefore, $I_r = 100G/G_{ref}$, where G and G_{ref} are solar irradiation (w/m^2) and that of under the standard test condition: $G_{ref} = 1000(w/m^2)$, respectively. The range of I_r is $I_r \in [0, 100]$. An example of the output voltage and current relationship is shown in **Figure 2**. We can see that the solar panel is a type of current sources, but the current is

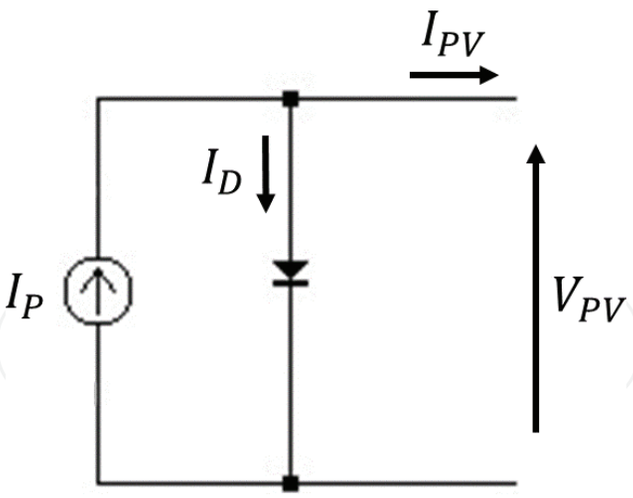


Figure 1. Equivalent circuit of a photovoltaic.

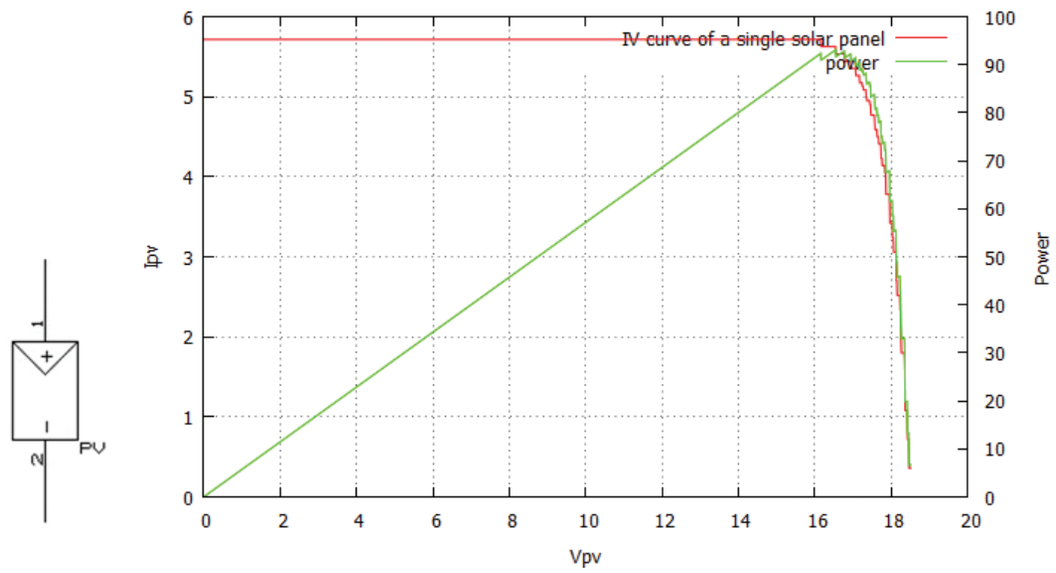


Figure 2. Single solar panel property. (Ir $= 80\%$, $N_p = 4$, $N_s = 12$, $I_{sc} = 1.8[A]$, $T = 298.15$, $q = 1.6 \times 10^{-19}$).

reduced when the voltage is higher than a certain value. The solar panel does not pass current if the panel is covered by a shadow. If series-connected solar panels have a partial shadow, the output current from the solar panels are down to zero even if a part of solar panels do not have a shadow. To prevent such a situation, a bypass diode is connected to each solar panel in parallel. Using this circuit, the solar panels can generate a certain amount of electricity even if they are partially shadowed. Such series-connected solar panels, however, show highly nonlinear characteristics (see **Figure 3**).

To extract maximum power, the voltage of the photovoltaic should be maximized. However, if the voltage is too high, the current decreases. Therefore, there is an optimal voltage value that maximizes the power. Such voltage is called the MPP and the power conditioner or converter connected to the PV tracks the MPP.

http://dx.doi.org/10.5772/intechopen.79711

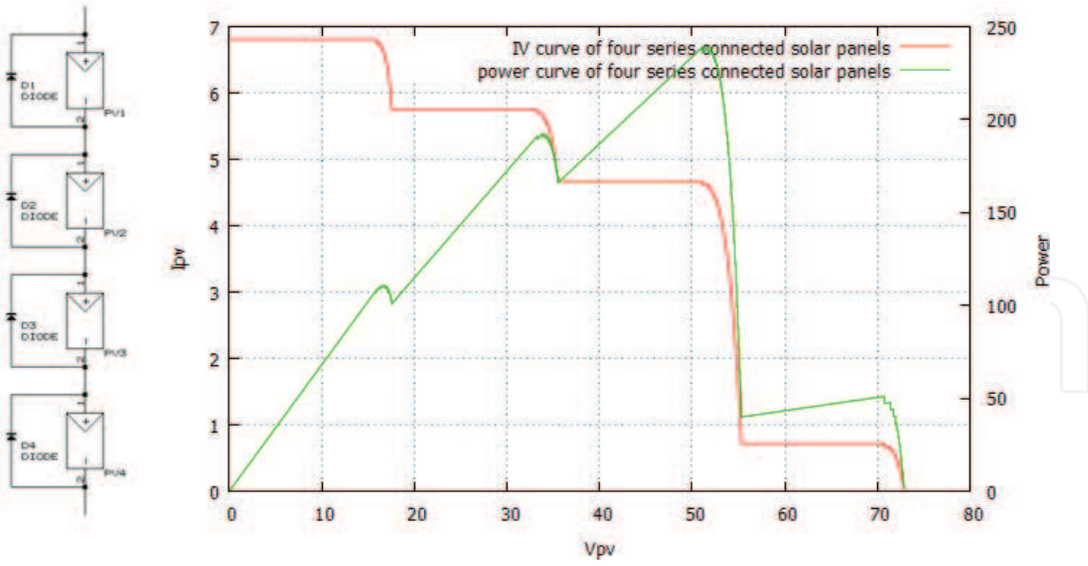


Figure 3. An example of series-connected solar panel property. The irradiations for the four panels are 10, 80, 65, and 95%.

Another noticeable property is that the current flow of photovoltaics stops when it has a shadow. Thus, if a photovoltaic is connected to the other photovoltaics in series and it has a shadow, no power is outputted from the series-connected solar panels.

This problem is solved by connecting a bypass diode in parallel with each photovoltaic. Using this architecture, we can get some amount of power even if a part of the solar panels are under a shadow. However, in such a case, the voltage-power curve of the photovoltaics shows a nonlinear form. As the voltage-power curve has several peaks, the power conditioner cannot obtain the correct MPP only using a hill-climbing technique. The most reliable method to solve this problem is for the power conditioner/DC converter to acquire the current voltage-power curve and detect the global maximum point.

3. MPPT algorithm accelerated by learning machines

One way to realize a quick MPPT without involving any special device is to use a photovoltaic model to predict the MPP. Moreover, the apparent property of photovoltaic varies due to the accumulated dust on the solar panel surfaces. This means that the photovoltaic model is not stable, but is valid depending on the solar panel's situation. To adjust to such changes in the property, an on-site learning machine should learn the MPP acquired by the P&O method to construct the PV model and apply prediction using the learning machine. In our previous work [7], we demonstrated that an incremental learning method on a budget on a microcomputer can manage the learning and prediction of MPPs. The learned results were applied only when solar irradiation changes drastically and the learning machine know the appropriate MPP that fits the current situation.

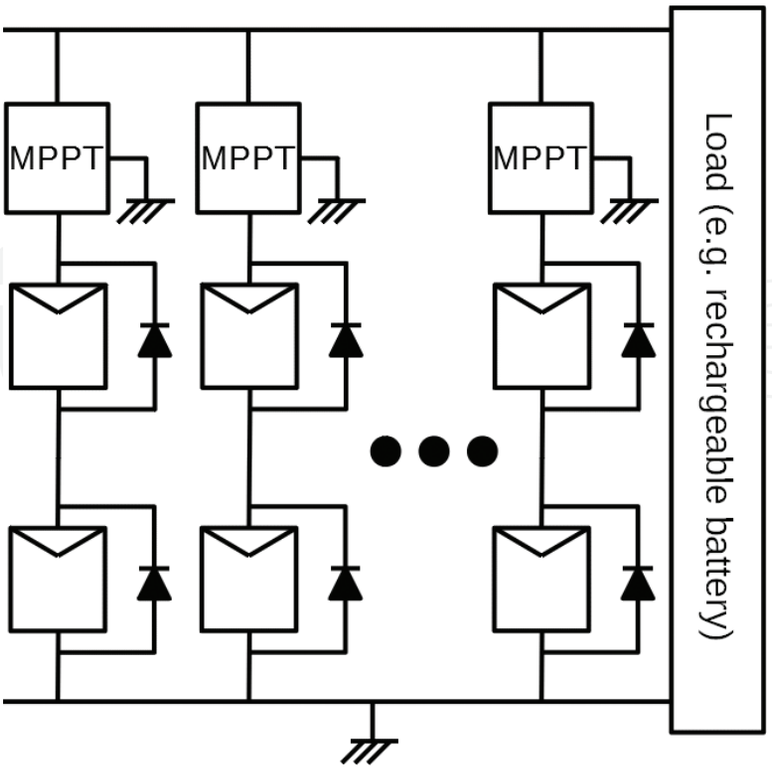


Figure 4. The photovoltaic circuit design bedded on road. Several solar panel strings with the MPPT converter are connected in parallel.

The previous system, however, cannot support the MPPT for series-connected PVs with bypass diodes, as shown in **Figure 4**. This is because even if the strength of solar irradiation is a certain stable value, there are several different solutions depending on the variety of the shadow patterns on the solar panels. To overcome this difficulty, we propose a new MPPT method in this chapter that is based on modal regression on a budget, which is a modal regression with a fixed number of kernels. Modal regression has the ability to approximate multivalued functions. Modal regression on a budget continues the learning with a fixed number of kernels so that it is suitable to be embedded in a small microcomputer. Therefore, it is able to record several different MPPs corresponding to the strength of solar irradiation. The proposed MPPT has a modified P&O method that enables tracking of MPPs from the voltage-power curve having several peaks using modal regression on a budget.

During the service, the proposed MPPT tracks the peaks by changing the initial search points. If an MPP is observed, the kernel density estimator (KDE) in the modal regression records the peak by adding a new kernel that records the current peak (see **Figure 5**). However, the microcomputer has limited storage space. Thus, if the number of kernels in the KDE equals the budget, one of the existing kernels will be replaced by the new kernel.

3.1. A perturbation and observation (P&O) method with changing initial point

Even if the system uses modal regression, it cannot be used before learning. Thus, it needs to obtain the MPPs first. To find several peaks, a modified P&O method is presented. The modified

http://dx.doi.org/10.5772/intechopen.79711

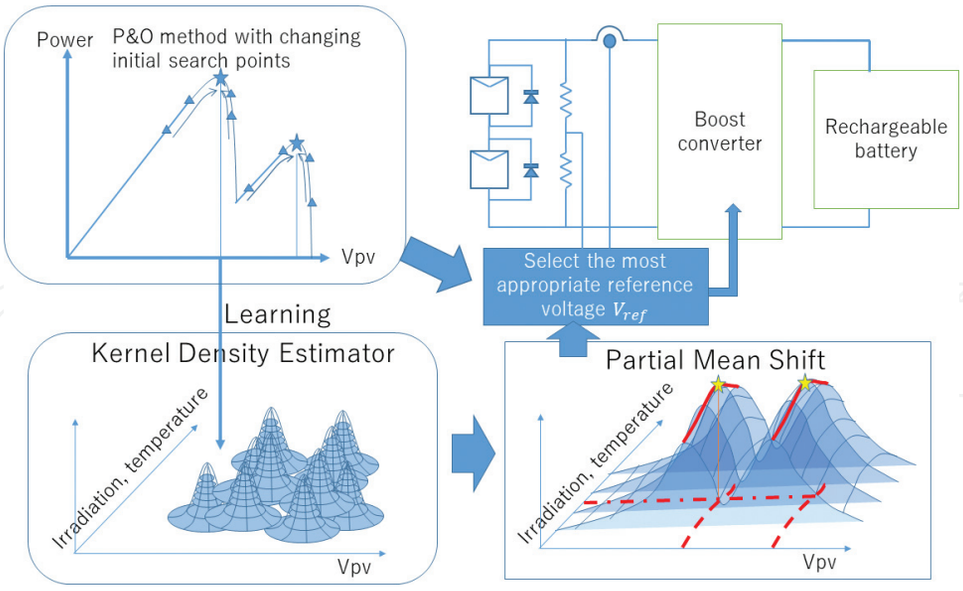


Figure 5. Outline of the MPPT accelerated by the modal regression on a budget.

one searches the peak points roughly at first. For example, if the solar panel comprises m number of clusters, the number of peaks would be up to m. Therefore, the new P&O estimates following C $(> m)$ initial points. This operation concludes when irradiation is greatly changed.

$$v_n^i = n \frac{V_{pv}^{max}}{C}, where\ n = 1, \cdots, C, \tag{2}$$

where V_{pv}^{max} denotes the open circuit voltage of the photovoltaic. To obtain this value, the circuit should be opened for a while when the irradiation changes. The system finds the $v_{n^*}^i$ that leads to maximum MPP.

$$n^* = argmax_n \left\{ P_{pv}\left(v_n^i\right) \right\}, \tag{3}$$

where $P_{pv}\left(v_n^i\right)$ denotes the power from the solar panel for the voltage v_n^i. In this method, m number of clusters are needed to be preset.

3.2. Modal regression on a budget for reasoning from too less sensory inputs

In general, if the device has too few sensors, the system cannot properly detect the current status. The partial shadow problem is one such problem. Therefore, if the device has illuminance sensors for each solar cell, it can accurately detect the status and can form complete relationships between the large number of sensory inputs and MPP. However, such strategy is impractical for real applications. Moreover, we should reduce the number of dimensions to construct an insect's brain like compact learning machine. From a theoretical viewpoint, the system having too few sensory inputs should yield several possible solutions. Therefore, the system has to check the suitability of all possible solutions and choose the best solution. One way to solve this problem is to employ a quick search

algorithm such as the PSO algorithm. However, PSO searches possible solutions for arbitrary initial setting of particles and wasted some time for the search. An alternative way to speed up the procedure is by implementing a learning machine to quickly obtain some good solution candidates. However, to realize such tasks, the learning machine has to have an ability to approximate multivalued functions. Such ability cannot be served by normal regression methods.

Modal regression approximates a multivalued function to search the local peaks of a given sample distribution. Modal regression comprises the KDE with a partial mean shift (PMS) method. We have already presented a minimum modal regression, which minimizes the number of kernels for the modal regression [12].

The model, however, does not support learning on a fixed budget. In this chapter, we propose an improved version of our previous work, which enables learning on a fixed budget.

3.2.1. Original modal regression method

Modal regression comprises KDE followed by the PMS. KDE is a variation of the Parzen window [13]. Let $\aleph$ be the set of learning samples and $\aleph = \{\boldsymbol{x}_p \in \Re^N | p = 1, 2, \ldots N\}$. The estimator approximates the probability density function using a number of kernels, namely the support set S_t. The kernels used are Gaussian kernels and

$$p(\boldsymbol{x}) \propto \sum_{i \in S_t} K\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|}{h_x}\right) \tag{4}$$

where

$$K\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|}{h_x}\right) \equiv exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{h_x^2}\right) \tag{5}$$

Normally, the same number of kernels as that of the dataset is required. However, if the storage capacity of a target device is small, the number of kernels must be restricted. There are several ways to realize density estimation using a limited number of kernels. Traditionally, self-organizing feature maps or learning vector quantization methods approximate the distribution using a fixed number of templates.

As mentioned in a previous study [14], the KDE used in modal regression should approximate the number of peak points of the distribution, rather than the distribution itself. Let $\widehat{p}(\mathbf{x})$ be

$$\widehat{p}(\boldsymbol{x}) \equiv \sum_{i \in S_t} K\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|}{h_x}\right) \tag{6}$$

then $\widehat{p}(\mathbf{x})$ should satisfy the following condition.

http://dx.doi.org/10.5772/intechopen.79711

$$\begin{cases} \nabla_x \widehat{p}(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}^*} = \nabla_x p(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}^*} = 0 \\ \nabla_x^2 \widehat{p}(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}^*} < 0, \quad \nabla_x^2 p(\boldsymbol{x})\Big|_{\boldsymbol{x}=\boldsymbol{x}^*} < 0 \end{cases}. \tag{7}$$

where $\mathbf{x}^*$ denotes a local peak point of the distribution.

Modal regression searches the peaks of the distribution model represented by the KDE. The PMS method realizes quick convergence to the nearest peak from the initial point. Let us denote the initial point as $\mathbf{x}_0$, representing the starting point for searching the peaks. Thus, modal regression repeats the modification of the current y as follows:

$$y_{new} \leftarrow \frac{\sum_i y_{old} K\left(\frac{|y_{old}-y_i|}{h_y}\right) K\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|}{h_x}\right)}{\sum_j K\left(\frac{|y_{old}-y_j|}{h_y}\right) K\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|}{h_x}\right)} \tag{8}$$

3.2.2. Modal regression on a fixed budget

To embed the modal regression, we have to pay attention to how to reduce the number of kernels for the KDE. Especially, we have to fix the upper bound for the number of kernels. In this case, the aim of the KDE is to approximate the peaks in the distribution rather than approximating the distribution. From this viewpoint, we should prune redundant kernels that do not contribute to approximating the peaks.

In our previous work [12], we demonstrated that the kernel, which is linearly dependent on the other kernels, can be removed without changing existing peaks. To this end, before pruning, the pruned kernel should be projected to the space spanned by the other remaining kernels. However, preparing the gram matrix wastes huge memory space.

Moreover, in this practical application, we should pay attention to the concept drift phenomena, wherein the labels change over time. This is caused by environmental changes such as the accumulation of dust on the solar panels and the changes in properties of the solar panel materials. The learning methods should support these issues.

To overcome these difficulties, we propose a simplified version of the modal regression method on a fixed number of kernels.

To discuss the learning rule of the KDE, let us rewrite the kernel output value as the dot product of the two vectors of $\mathrm{k}(\boldsymbol{x}_\mathrm{i}, .)$ and $\mathrm{k}(\boldsymbol{x}, .)$ as follows.

$$\langle k(\boldsymbol{x}_i, \cdot), k(\boldsymbol{x}, \cdot)\rangle \equiv K\left(\frac{\|\boldsymbol{x}-\boldsymbol{x}_i\|}{h_x}\right), \tag{9}$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product operator. This expression is based on the kernel method. Fortunately, the Gaussian kernel is a type of reproducing kernel in which we can rewrite the learning rule using the dot product of vectors. Using this representation, we can rewrite the

learning rule in algebraic expressions, which can be very easily understood. Now, let us denote a vector $\widehat{P}_t$ as the learning result after the t-th sample presentation. Then, we obtain

$$\widehat{P}_{t-1} \equiv \sum_{i \in S_{t-1}} W_i k(\boldsymbol{x_i}, \cdot), \tag{10}$$

where S_{t-1} denotes the support set after the $t-1$-th presentation of a given sample. The KDE output to an input vector $\boldsymbol{x}$ is calculated by

$$\widehat{P}_{t-1}(\boldsymbol{x}) = \left\langle \widehat{P}_{t-1}, k(\boldsymbol{x}, \cdot) \right\rangle. \tag{11}$$

Eq. (10) enables us to represent the learning rule as

$$\widehat{P}_t = \widehat{P}_{t-1} + y_t k(\boldsymbol{x}_t, \cdot), \;\; S_t = S_{t-1} \cup \{t\} \tag{12}$$

However, the proposed method restricts the number of kernels to a certain number as $|S_t| \leq B$. To overcome this problem, the proposed method replaces one of the kernels with a new kernel whose centroid is the new input vector, or moves the nearest kernel centroid to close to the current new input vector. Therefore, if the nearest kernel

$$n_t = argmin_j \left\{ \left\| \boldsymbol{x_t} - \boldsymbol{x_j} \right\|^2 \right\}, \tag{13}$$

satisfies the following condition

$$\| \boldsymbol{x_t} - \boldsymbol{x}_{n_t} \|^2 < \theta_{activity}, \tag{14}$$

its kernel center is modified to be the mean vector of the original kernel center and the new sample as follows. The extension coefficient W_{n_t} is increased by Δ.

$$\mathbf{x}_{\mathrm{n_t}} = \frac{\left(\frac{\mathrm{W_{n_t}}}{\Delta}\right)\mathbf{x}_{\mathrm{n_t}} + \mathbf{x}_{\mathrm{t}}}{\left(\frac{\mathrm{W_{n_t}}}{\Delta}\right) + 1}, \;\; \mathrm{W_{n_t}} = \mathrm{W_{n_t}} + \Delta \tag{15}$$

The extension coefficient includes information on how many samples did the kernel learn. The extension coefficient is also reflected to a weighted PMS method in Eq. (20). However, if the kernel center does not satisfy the Eq. (14), one of the kernels should be replaced with the new tentative kernel. Therefore, if the new sample $\mathbf{x_t}$ is too far from the nearest kernel center, one of the kernels should be replaced with it to adjust to the new sample. In such a case, the least recently or frequently used (LRFU) kernel is to be replaced with the new one. The LRFU evaluation method proposed in [15] is an improved version of the LRU page-replacement algorithm for virtual memory systems on operating systems. Using this evaluation method, the most ineffective kernel, which seems to be unused for a long time interval, is replaced with the new kernel. To realize this evaluation, a variable that represents the value of each kernel is introduced. Let C_i be the value of the i-th kernel. When the i-th kernel centroid is the closest to current sample x_t, C_i is enlarged, but is decreased, otherwise. Therefore, for each round, the following equation is executed.

http://dx.doi.org/10.5772/intechopen.79711

$$C_i = \begin{cases} C_i + 1, & i = n_t \\ \eta C_i, & i \neq n_t \end{cases}, \tag{16}$$

where $\eta = 1 - \epsilon$, $\epsilon \ll 1$. Then, the $j^* = argmin_i C_i$ kernel is to be replaced with the new kernel. Therefore,

$$\boldsymbol{x_{j^*}} = \boldsymbol{x_t}, \;\; w_{j^*} = \Delta, \;\; C_{j^*} = 1, \;\; y_{j^*} = y_t \tag{17}$$

h_x determines the width of each kernel. The performance of the system is also sensitive to this value, so we have to set this value carefully. In a previous study [16], the optimal value of h_x for a standard distribution was derived as

$$\mathrm{h_x} = \left(\frac{4}{d+2}\right)^{\frac{1}{d+4}} n^{-\frac{1}{d+4}}, \tag{18}$$

where $\mathrm{d} = \dim(\boldsymbol{x_t})$ is the dimension of the input vector and n is the number of samples. In this study, the number of samples is unknown. However, the number of kernels are bounded to the budget B so that $\mathrm{n} = \mathrm{B}$. Equation (18), however, cannot be used for practical applications. Therefore, we should consider a scaling factor for (18). To this end, in this study, we rewrite (18) as follows.

$$\mathrm{h_x} = v_0 \cdot \left(\frac{4}{d+2}\right)^{\frac{1}{d+4}} n^{-\frac{1}{d+4}}, \tag{19}$$

where v_0 denotes the scaling factor and was set to 0.3 in this simulation described in Section 4. Actually, in the simulation described in Section 4, each input dimension was normalized before the execution of the modal regression. Concretely, each element of $\boldsymbol{x_t}$ of modal regressor was multiplied by a gain g_i to make the range of the ith element of $\boldsymbol{x_t}$ be $\left|g_i x_{ti}\right| \leq 1$. The output from the modal regressor (20) was divided by the corresponding gain $y = y/g_o$. For simplicity, however, following text omit the description of these gains.

The regression output is also delivered by the PMS method described in (8). In this model, the PMS method should account for the extension parameter $\mathrm{W_i}$. To this end, this method also uses the weighted PMS method as is done in our previous work [12]. Note that (20) includes the extension parameter $\mathrm{W_i}$ in both the numerator and the denominator.

$$y_{new} \leftarrow \frac{\sum_i y_{old} W_i K\left(\frac{\left|y_{old} - y_i\right|}{h_y}\right) K\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|}{h_x}\right)}{\sum_j W_j K\left(\frac{\left|y_{old} - y_j\right|}{h_y}\right) K\left(\frac{\|\boldsymbol{x} - \boldsymbol{x}_j\|}{h_x}\right)} \tag{20}$$

The weighted PMS should be repeated by substituting derived $\mathrm{y_{new}}$ to $\mathrm{y_{old}}$ until it converges to a certain value. In the computer simulation described in Section 4, the weighted PMS was repeated 10 times for every initial point. This process is executed for all initial values of y_{old} to obtain all local peaks. The simplest way to set the initial points is choosing uniform random initial values for $\mathrm{y_{old}}$. However, the random initial values usually make some unexpected

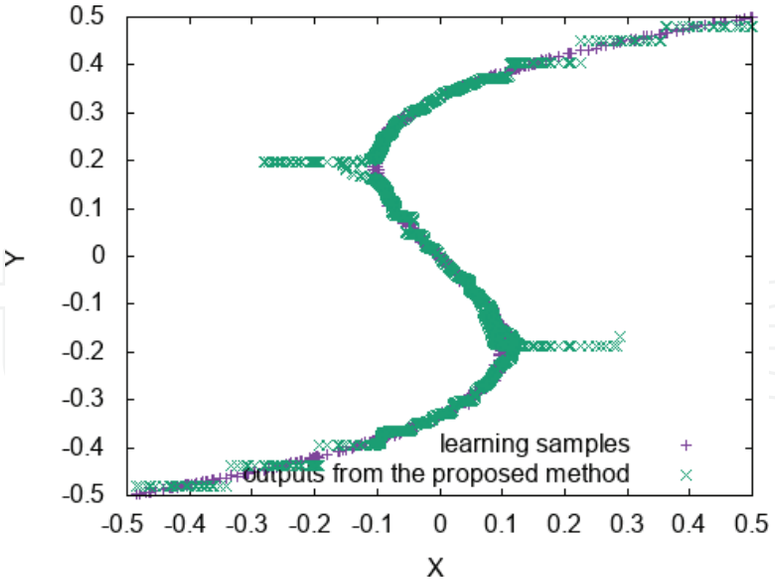


Figure 6. The response for the third-order data. The green curve is the response of the proposed model with 50 kernels.

converged values for y. To more appropriately set up the initial value y_0, the proposed method chooses the initial value as the corresponding element of each kernel center. Therefore, let us assume that a kernel center $\mathbf{x}_i$ is similar to the current input. Then, the initial value should be $y_0 = x_{ij}$, where j is the corresponding unknown dimension. The set of such kernel centers is

$$S_{active} \equiv \left\{ i | exp\left(\sum_{j \neq unknown} \frac{-(x_{ij} - x_{tj})^2}{h_x^2} \right) > \theta_{init} \right\}, \tag{21}$$

where θ_{init} denotes the threshold for choosing the kernel. The above equation does not contain the distance calculation for the unknown dimension. The initial values for y_0 are

$$y_0 = x_{k\ unknown} \quad where\ k \in S_{init}. \tag{22}$$

3.2.3. An example of the modal regression outputs

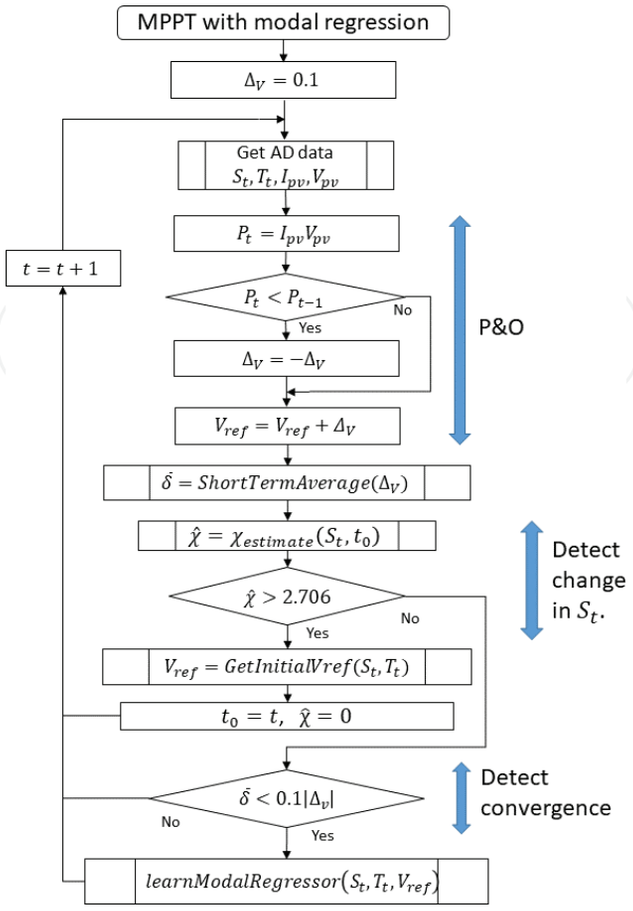
The modal regression approximates multivalued functions. As an example, **Figure 6** shows the regression output for 800 sets of third-order synthetic data with 50 kernels. We can observe that the proposed method partly approximates multivalued function.

3.3. Whole algorithm

Algorithms 1–4 are presented below. Note that S_t in these algorithms shows the averaged solar irradiation for all clusters. Therefore, solar irradiation is assumed to be sensed by a single illuminance sensor; thus, the obtained value is the average of the values of both clusters.

The algorithm is roughly divided into two parts: one is the normal P&O part, and the other deals with searching for the reference voltage using the proposed modal regression. The second part is executed when the solar irradiation is changed abruptly.

http://dx.doi.org/10.5772/intechopen.79711

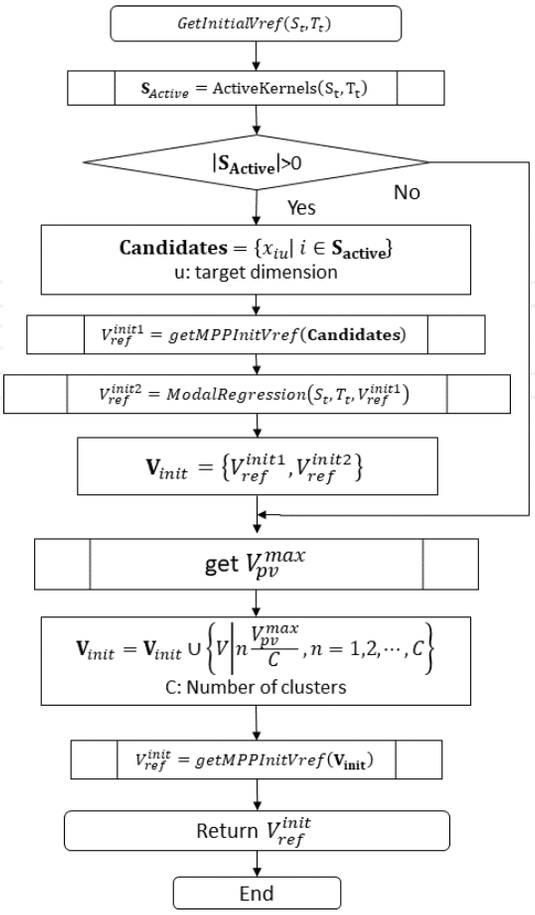


Algorithm 1. Algorithm of the MPPT with modal regression. Note that V_{ref} is referenced by the proportional-integral-derivative (PID) control thread at each time interval. *GetInitialVref()* is described in **Algorithm 2**. *learnModalRegressor*() is described in Eqs. (12)–(19).

To this end, the sensed solar irradiation is statistically analyzed by $\chi_{\mathbf{estimate}}(S_t, t_0)$, that is chi-square test for $\mathbf{S_t}$ from the previously changed time $\mathbf{t_0}$ till now. If it includes an obvious change, **Algorithm 2** is called to search for an appropriate initial V_{ref}'s for searching the optimal value of V_{ref}. **Algorithm 2** is the algorithm for searching initial V_{ref}. This algorithm conducts a search using the proposed modal regressor followed by a search of the initial V_{ref} using the proposed modified P&O algorithm described by Eq. (2). The reason why it executes an additional search is that there is a possibility that the modal regressor yields incomplete solution candidates. Such a situation usually occurs when the modal regressor is in the early stage of learning.

3.4. Computational cost and required memory capacity

The computational cost for the MPPT with modal regression is mainly wasted by the modal regressor. Hence, let us consider the computational cost for the modal regression. Now, we assume that the number of kernels in the modal regressor is B and that the number of dimensions is N. Note that N=**3** because input vector is $x_t = [S_t, T_t, V_{ref}]^T$. To calculate the kernel outputs for current input x_t, it needs (**N**+**1**)B times multiplies and B times of division and B times of calculation of **exp** (). If we assume that the calculation of **exp** () is C_{exp}, the



Algorithm 2. Pseudo code for getting initial reference voltage. *ActiveKernels*() is derived by (21). *getMPPInitVref*() is described in **Algorithm 3**. *ModalRegression*() is the five time repeats of the partial mean shift:(8).

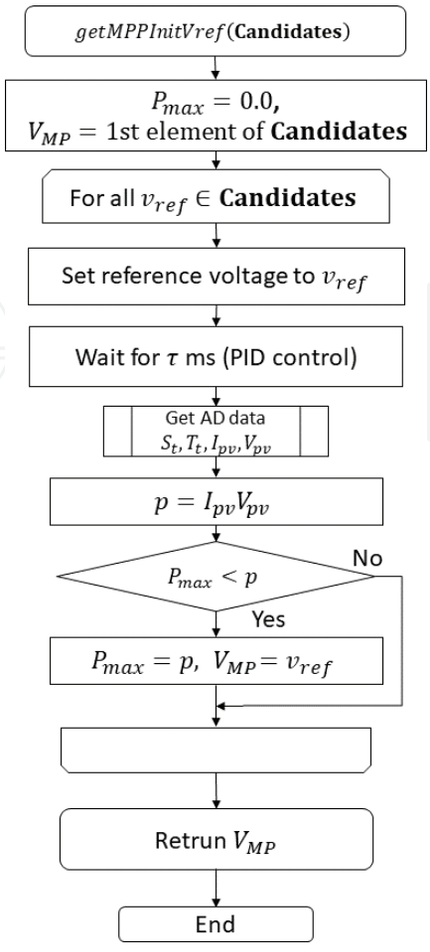
computational cost is proportional to $\boldsymbol{B}(\boldsymbol{N}+\mathbf{2}+\boldsymbol{C}_{exp})$. Therefore, to derivate a kernel set $\boldsymbol{S}_{active}$ in (21), it needs $\boldsymbol{O}(\boldsymbol{B})$. The partial mean shift (20) needs $\boldsymbol{B}(\boldsymbol{N}+\mathbf{8}+\mathbf{2}\boldsymbol{C}_{exp})+\boldsymbol{B}(\boldsymbol{N}+\mathbf{7}+\mathbf{2}\boldsymbol{C}_{exp})+\mathbf{1}=\boldsymbol{B}(\mathbf{2}\boldsymbol{N}+\mathbf{15}+\mathbf{4}\boldsymbol{C}_{exp})+\mathbf{1}$. Thus, if the partial mean shift is repeated for $\boldsymbol{M}$ times for each trial, the total computational power of modal regression is proportional to $\boldsymbol{MB}(\mathbf{2}\boldsymbol{N}+\mathbf{15}+\mathbf{4}\boldsymbol{C}_{exp})+\mathbf{1}$.

The computational power required for the learning of the modal-regressor is the cost of executing (13), (14), and (16). Thus, it needs $\boldsymbol{BN}+\boldsymbol{N}+(\mathbf{2}\boldsymbol{N}+\mathbf{1})$ multiplications. After all, the computational complexity of the modal regression is $\boldsymbol{O}(\boldsymbol{B})$.

The required memory capacity also depends on the number of kernels. Each kernel records the center of kernel $\boldsymbol{x}_i$, corresponding label $\boldsymbol{y}_i$, the extension parameter $\boldsymbol{W}_i$ and the parameter $\boldsymbol{C}_i$ for the LRFU estimation. As each float variable requires 4 bytes, one kernel requires $\mathbf{4}(\boldsymbol{N}+\mathbf{2})$ bytes. Thus, the total amount of memory storage for all kernels is $\mathbf{4}\boldsymbol{B}(\boldsymbol{N}+\mathbf{2})$ bytes.

The boost converter step ups the voltage of the solar panel string and charges the battery. The MPPT unit, which includes the proposed method, sends the predicted MPP: V_{ref} to the feedback controller. The P-type MOSFET is assumed to be used for making an open circuit in a short-time interval to get V_{pv}^{max} (see **Algorithm 2**). As shown in **Figure 4**, several sets of this circuit are connected in parallel to the same rechargeable battery.

http://dx.doi.org/10.5772/intechopen.79711



Algorithm 3. Flowchart for *getMPPInitVref*().

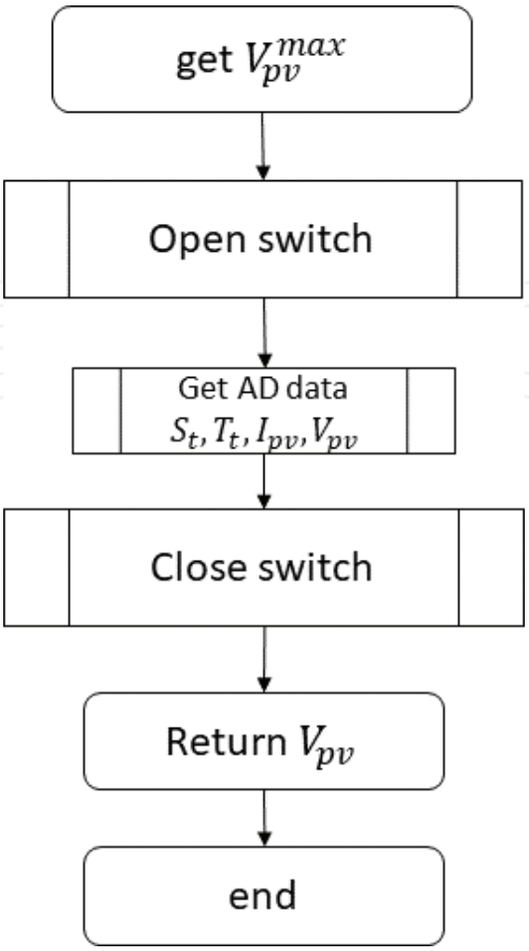
4. Computer simulation

The performance of the proposed MPPT was evaluated via a simulation. Particularly, the convergence speed to MPP is a very important property that should be evaluated. The simulated circuit comprises a short string of solar panels connected to a boost converter (see **Figure 7**).

The MPPT unit sends the reference voltage V_{ref} for the feedback controller, and the boost chopper circuit adjusts the output voltage of the PV string to V_{ref}. In this simulation, we assume that the load is a rechargeable battery, whose voltage is kept to a certain constant voltage. Using this load, each boost converter is not affected by the change in the other converter's output power.

For simplicity, the simulator of the boost converter simply updates V_{pv} to be V_{ref} and calculates the corresponding I_{pv} by using the photovoltaic model. Therefore, the detailed transient response of the boost converter was not realized in the simulator.

To realize the simulation, we constructed a simulator of photovoltaics and circuits as the Java application. The solar irradiation, temperature, and the properties of the solar panels are also represented in the thread of environment class (see **Figure 8**).



Algorithm 4. Flowchart for getting V_{pv}^{max}. Open and close switch denote enabling and disabling the FET in **Figure 7**.

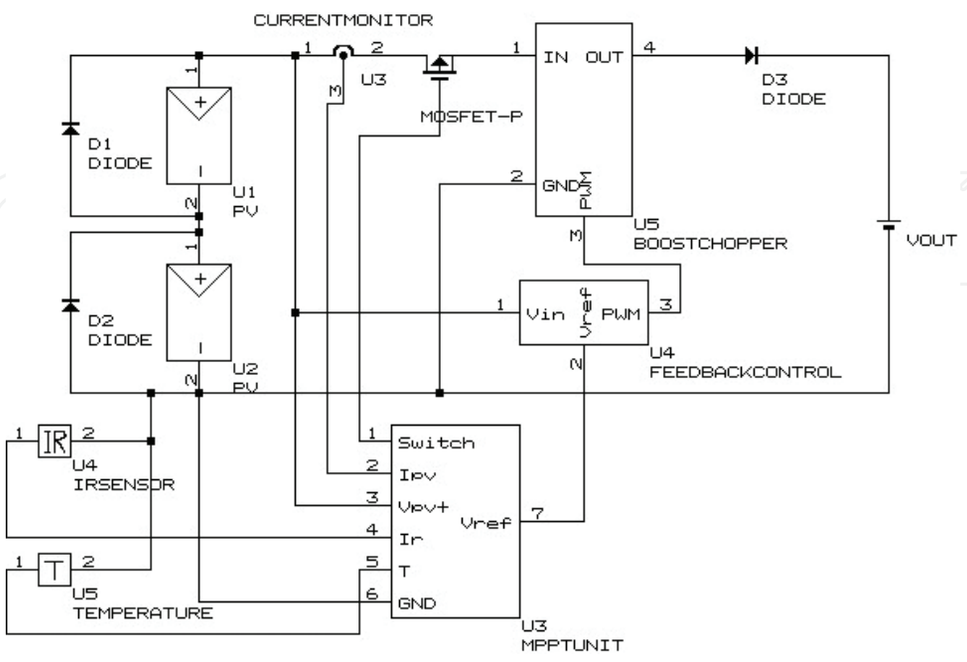


Figure 7. The circuit for the simulation.

http://dx.doi.org/10.5772/intechopen.79711

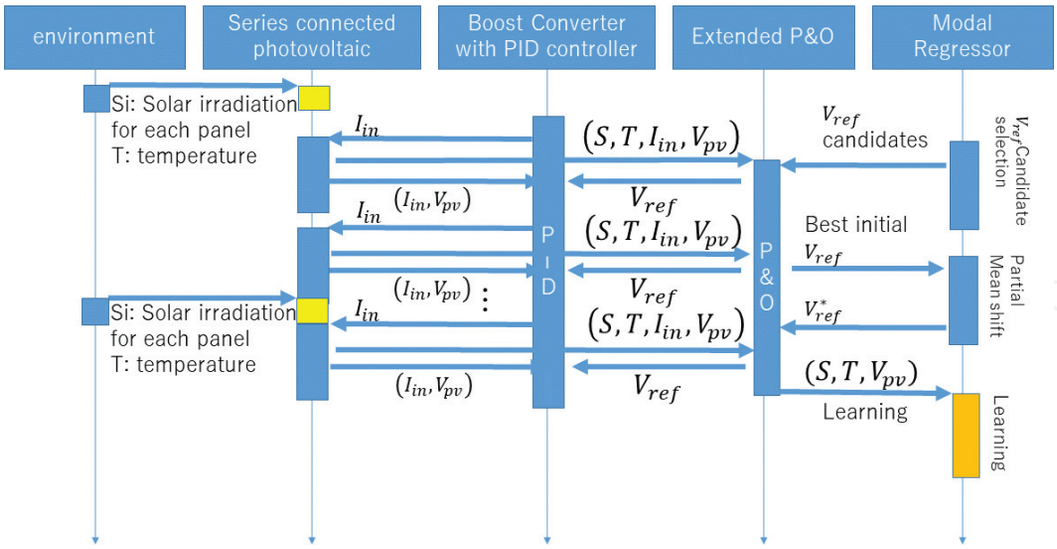


Figure 8. Sequence diagram of the simulator.

For simplicity, the strength of solar irradiation and temperatures varies for a certain scenario, but the effect of the specific heat of the solar panel material was not considered.

The solar panel is a homogeneous two cluster panel such that it has two peaks under partial shadow conditions. The MPPT with modal regression is also represented by the MPPT thread class. The chopper circuit with the feedback controller is assumed to control the output voltage from the solar panel to $\boldsymbol{V_{ref}}$, which is assigned by the MPPT unit, within 1 ms. Note that $\boldsymbol{V_{ref}}$ is yielded by the modified P&O method or the modal repressor. Similar to the simulation method proposed in [10], the chopper circuit is simulated so as to change $\boldsymbol{I_{pv}}$. As a result, the series-connected solar panel simulator yields a new $\boldsymbol{V_{pv}}$ due to the change in $\boldsymbol{I_{pv}}$. The new $\boldsymbol{V_{pv}}$ is then sent to the boost converter simulator to calculate the next step.

We have compared the proposed method with the existing models under partial shadow conditions. For this comparison, the following three models were prepared: MPPT with the modal regression, the P&O method by changing initial points described in Section 3.1, and MPPT with PSO. There are various PSO-based MPPT methods [8, 17]. In this simulation, we prepared a model that is based on the model proposed in [17] because it has a similar

Δ_v: Change in voltage for P&O (**Algorithm 1**)	0.1
θ_{init} in Eq. (21)	0.9
η in Eq. (16)	0.001
d in Eq. (19)	3
Time interval for changing V_{ref} by P&O, modal regression and PSO (=τ in **Algorithm 3**)	1 [ms]
Time interval for changing solar irradiation	250 [ms]
Scaling factor v_0 in (19)	0.3
Number clusters (= C in **Algorithm 2**). This value should be greater than the actual number of clusters.	3

Table 1. Parameters used in this simulation.

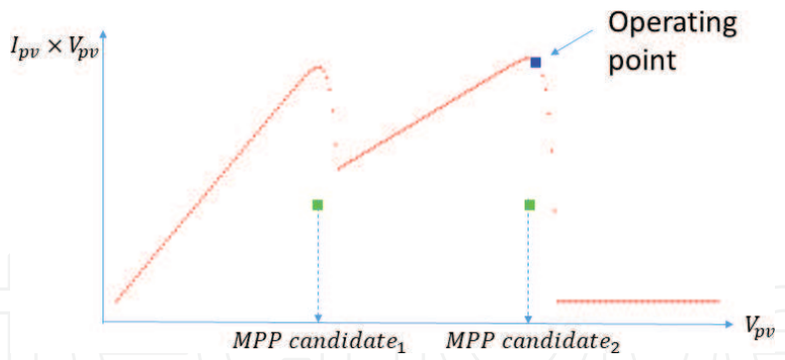


Figure 9. An example of snapshot of the maximum power tracking of the proposed method. The green points are the center points of the proposed modal regressor, namely the initial MPP candidates (see (21)).

architecture to ours. The PSO-based MPPT method used in this simulation executes the PSO optimization when solar irradiation changes is occurred. The condition for detecting solar irradiation changes was the same as the method described in Section 3.3. The detailed parameters used in this simulation are listed in **Table 1**.

We evaluated the electric power generation behavior of each model. If the generated power is higher than the others, the model finds MPP faster than the others.

Figure 9 shows a snapshot of the behavior of our proposed MPPT. In this situation, the power-voltage curve of the solar panel has two peak points. The activated kernel centers of the modal-regression at this situation are shown as the two green points[1]. The proposed method set choose one of them as the start point for the MPPT. After that, the modal regression output was used for the

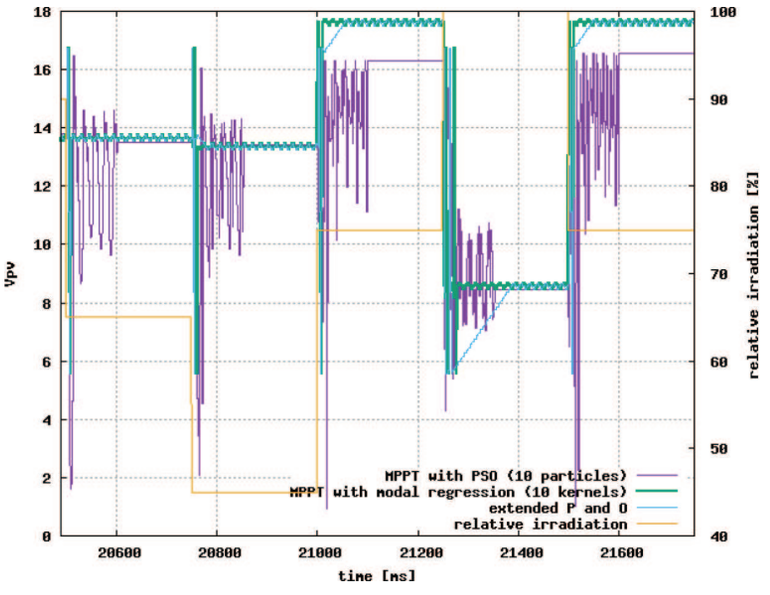


Figure 10. An example of V_{pv} VS time.

[1]The activated kernel centroids without the power element were pointed as the green points. However, the height of the green points have been set to a certain fixed value for easy seeing.

http://dx.doi.org/10.5772/intechopen.79711

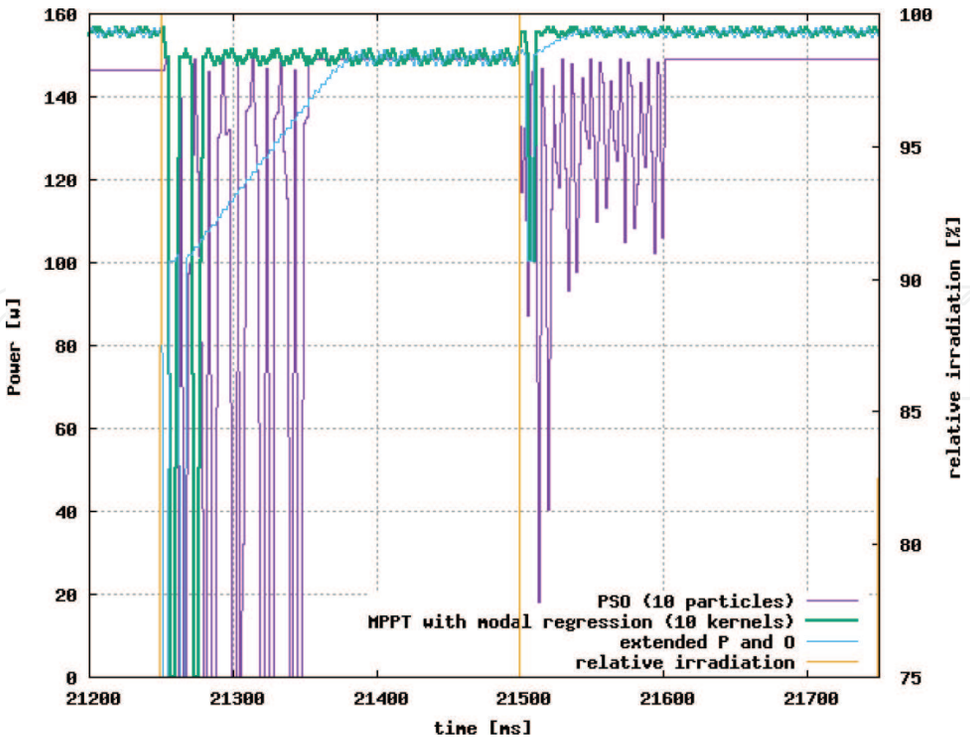


Figure 11. The magnified power curves. Note that the power curve has changed immediately after the change of irradiation.

initial point for starting the P&O procedure. As a result, the proposed method finds the MPP faster than the P&O method. The quick search ability is suitable for generating electricity under changing irradiation. In **Figure 10**, the green, blue, and purple curves show the V_{pv} of the proposed one, P&O-, and PSO-based MPPT methods, respectively. The V_{pv} of the PSO-based method changes drastically for approximately 100 ms immediately after the change in solar irradiation. Although the proposed and P&O methods also change V_{pv} immediately after the change in solar irradiation, the changing period is shorter than that of the PSO-based method. Moreover, V_{pv} of P&O-based method sometimes needs a time interval to converge to be a steady state. On the contrary, the proposed method makes V_{pv} reach the steady state faster than the others.

The magnified **Figure 11** shows that the power generation of our proposed method quickly aliased immediately after the change in solar irradiation, whereas the extended P&O method gradually converges to the power of the proposed method. The PSO-based MPPT shows the less power generation than the other methods. In the case of PSO, the results are greatly affected by the initial points of the particles. In this simulation, we have set the initial points by uniform random voltages in $\left[0, V_{PV}^{max}\right]$, where V_{PV}^{MAX} is the open-circuit voltage of the solar panel string. The initial points should be distributed uniformly in the interval. However, if the number of particles is small due to the restriction of the device, the initial point distribution usually becomes to be an unbalanced distribution. As a result, the quality of the solution is degraded. To check the performances under the various sizes of kernels or particles, the averaged generated power for the proposed method with 5 and 10 kernels, and the PSO-based MPPT methods with 5 and 10 particles were compared. Moreover, the generated electricity power from the proposed method and the extended P&O

Time interval for changing solar insolation	Method	Averaged electricity power
250 ms	MPPT with modal regression (5 kernels)	151.2 W
	MPPT with modal regression (10 kernels)	151.9 W
	Extended P&O	150.3 W
	MPPT with PSO (5 particles)	127.1 W
	MPPT with PSO (10 particles)	128.2 W
200 ms	MPPT with modal regression (10 kernels)	151.4 W
	Extended P&O	149.4 W

Table 2. Comparison of averaged electricity power generated during the first 200 [s]. The time interval for solar irradiation change were 250 and 200 [ms].

methods were compared with two different time intervals of changing solar irradiation. **Table 2** shows the results. We can see that the averaged generated power of the proposed method of 5 and 10 kernels are almost the same. On the other hand, the PSO-based methods reduced the power if the size of particles is reduced. The proposed method's generated power was also larger than the extended P&O method because the convergence speed is higher than that of the P&O method. The difference in the generated power is caused by their different convergence speed. Therefore, if there are fewer changes in solar irradiation, the difference decreases because the convergence process does not occur. As evidence, **Table 2** shows that if the time interval of changing solar irradiation is 250 ms, the differences between the two averaged generated power was 1.6 W, whereas the difference was 2 W when the time interval is 200 ms.

5. Conclusion

In this chapter, we proposed a new MPPT method accelerated by modal regression on a budget, which approximates multivalued functions. The modal regression on a budget is a simplified version of our previously proposed method, namely limited modal regression [12].

The proposed MPPT method comprises an irradiation sensor, temperature sensor, and modal regression on a budget. We assume that the irradiation sensor gets the averaged strength of irradiation of all solar panels. In the case for MPPT of PV strings, the device has to obtain the highest local peak point from the several peak points in the voltage-power curve. Therefore, the MPPT device with the incomplete sensory input has to approximate a multivalued function between the sensory inputs and the MPP.

Normally, modal regression estimates provide sample distribution and yield local peak points that are related to the specified input.

The modal regression on a budget can approximate such relationships between the sensory inputs and the MPP's. The proposed MPPT method is a combination of modal regression on a budget and a modified (extended) P&O method. The modified P&O method obtains the MPPs even if there are several local peak points. The obtained MPPs are recorded in the modal regressor.

The proposed method was evaluated by computer simulation under partial shadow conditions. The simulation results suggest that the MPPT with modal regressor obtain an MPP faster than other existing methods such as the MPPT with PSO. This property is suitable for electricity generation using the solar panels bedded on roads.

Acknowledgements

This study is sponsored and supported by KYODO Corporation, Toyota-city, Aichi-ken.

Author details

Koichiro Yamauchi

Address all correspondence to: k_yamauchi@isc.chubu.ac.jp

Department of Computer Science, Chubu University, Kasugai-shi, Aichi, Japan

References

[1] Bendib B, Belmili H, Krim F. A survey of the most used mppt methods: Conventional and advanced algorithms applied for photovoltaic systems. Renewable and Sustainable Energy Reviews. 2015;**45**:637-648

[2] Esram T, Chapman PL. Comparison of photovoltaic array maximum power point tracking techniques. IEEE Transactions on Energy Conversion. 2007;**22**(2):439-449

[3] Boehringer AF. Self-adapting dc converter for solar spacecraft power supply. IEEE Transactions on Aerospace and Electronic Systems. 1968;**AES-4**(1):102-111

[4] Esram T, Kimball JW, Krein PT, Chapman PL, Midya P. Dynamic maximum power point tracking of photovoltaic arrays using ripple correlation control. IEEE Transactions on Power Electronics. 2006;**21**(5):1282-1291

[5] Veerachary M, Senjyu T, Uezato K. Neural-network-based maximum-power-point tracking of coupled-inductor interleaved-boost-converter-supplied pv system using fuzzy controller. IEEE Transactions on Industrial Electronics. 2003;**50**(4):749-758

[6] Akkaya R, Kulaksiz AA, Aydogdu O. Dsp implementation of a pv system with ga-mlp-nn based mppt controller supplying bldc motor drive. Energy Conversion and Management. 2007;**48**:210-218

[7] Yamauchi K. Incremental learning on a budget and its application to quick maximum power point tracking of photovoltaic systems. Journal of Advanced Computational Intelligence and Intelligent Informatics. 2014;**18**(4):682-696

[8] Liu C-L, Luo Y-F, Huang J-W, Liu Y-H. A pso-based mppt algorithm for photovoltaic systems subject to inhomogeneous insolation. In: The 6th International Conference on Soft Computing and Intelligent Systems. 2012. pp. 721-726

[9] Noguchi T, Togashi S, Nakamoto R. Short-current pulse-based maximum-power-point tracking method for multiple photovoltaic-and-converter module system. IEEE Transactions on Industrial Electronics. 2002;**49**(1):217-223

[10] Tan YT, Kirschen DS, Jenkins N. A model of pv generation suitable for stability analysis. IEEE Transactions on Energy Conversion. 2004;**19**(4):748-755

[11] Bellia H, Youcef R, Fatima M. A detailed modeling of photovoltaic module using matlab. NRIAG Journal of Astronomy and Geophysics. 2014;**3**:53-61

[12] Koichiro Y, Vanamala Narashimha B. Minimum modal regression. In Maria De Marsico, Gabriella Sanniti di Baja, and Ana Fred, editors, ICPRAM2018 7th International Conference on Pattern Recognition Applications and Methods, pages 448-455, 2018

[13] Parzen E. On estimation of a probability density function and mode. Annals of Mathematical Statistics. 1962;**33**(3):1065-1076

[14] Sasaki H, Ono Y, Sugiyama M. Modal regression via direct log-density derivative estimation. In: Hirose A, Ozawa S, Doya K, Ikeda K, Lee M, Liu D, editors. Neural Information Processing –23rd International Conference, ICONIP 2016–, Volume Part II. Springer-Verlag; 2016

[15] Lee D, Noh SH, Min SL, Choi J, Kim JH, Cho Y, Sang KC. Lrfu: A spectrum of policies that subsumes the least recently used and least frequently used policies. IEEE Transactions on Computers. 2001;**50**(12):1352-1361

[16] Silverman BW. Density Estimation for Statistics and Data Analysis. CRC Press; 1986

[17] Rajasekar N, Vysakh M, Thakur HV, Mohammed Azharuddin S, Muralidhar K, Paul D, Jacob B, Balasubramanian K, Sudhakar Babu T. Application of modified particle swarm optimization for maximum power point tracking under partial shading condition. Energy Procedia. 2014;**61**:2633-2639