A Distributed Approach to Maximum Power Point Tracking for Photovoltaic Submodule Differential Power Processing

Shibin Qin, *Student Member, IEEE*, Stanton T. Cady, *Student Member, IEEE*,
Alejandro D. Domínguez-García, *Member, IEEE*, and Robert Carl Nikolai Pilawa-Podgurski, *Member, IEEE*

***Abstract*—This paper presents the theory and implementation of a distributed algorithm for controlling differential power processing converters in photovoltaic (PV) applications. This distributed algorithm achieves *true* maximum power point tracking of series-connected PV submodules by relying only on local voltage measurements and *neighbor-to-neighbor* communication between the differential power converters. Compared to previous solutions, the proposed algorithm achieves reduced number of perturbations at each step and potentially faster tracking without adding extra hardware; all these features make this algorithm well-suited for long submodule strings. The formulation of the algorithm, discussion of its properties, as well as three case studies are presented. The performance of the distributed tracking algorithm has been verified via experiments, which yielded quantifiable improvements over other techniques that have been implemented in practice. Both simulations and hardware experiments have confirmed the effectiveness of the proposed distributed algorithm.**

I. Introduction

In photovoltaic (PV) energy systems, PV modules are often connected in series for increased string voltage. However, there is usually mismatch between the *I–V* characteristics of the series connected PV modules; this is typically the result of partial shading, manufacturing variability, and thermal gradients. Since all modules in a series string share the same current, the string output power may be limited by underperforming modules. A bypass diode is often connected in parallel with each PV module to mitigate this mismatch and prevent PV hot spotting, but the efficiency loss is still significant when only a central converter is used to perform MPPT on the PV string.

To address the mismatch problem, distributed power electronics architectures that perform module-level MPPT, or even submodule-level MPPT, have been proposed; the two most dominant architectures are DC optimizers [1]–[4], and microinverters [5]–[7], as shown in Fig. 1(a) and (b), respectively. The major limitation of these two solutions, however, is that the distributed converters are connected in series with the PV modules and must process the full power output of every module. Even with highly efficient distributed converters, the overall power losses of the system can still be significant.

Differential power processing (DPP) for PV applications has gained significant attention recently due to its substantial improvements over conventional solutions in terms of efficiency, reliability, and cost. The structure of a DPP-based system is shown in Fig. 1(c); an overview of the DPP concept can be found in [8]. In contrast to dc optimizers or microinverters, DPP converters, or DPPs, are configured in parallel with the PV string. The bulk power that is common to all PV modules is directly processed by the central inverter without any intermediate conversion. DPP converters only need to process the power difference between series-connected PV modules, which is often just a small fraction of the bulk power. This results in high efficiency of the system, small size, and low power ratings of the power electronics circuit components. Due to these advantages, DPP has been further extended to apply to submodule-level applications [9]–[13].

In order to compare the efficiency of a DPP-based system with microinverters or dc optimizers, we consider a system with ten PV modules (i.e., 30 PV submodules) as an example. The irradiances on the PV submodules are set randomly by drawing from a Gaussian distribution with a standard deviation equal to 10% of the mean; the resulting maximum power of each PV submodule is displayed in Fig. 2 using striped red bars. The differential power, i.e., the power difference between the maximum power of each PV module and the power common to all PV modules, is also shown in Fig. 2 using dotted blue bars. In a dc optimizer-based system, the dc optimizers have to process the full power of all PV submodules (represented by the striped red bars), which adds up to 2218 W. In a DPP-based system, each DPP only needs to process the differential power (represented by the dotted blue bars), which only adds up to 242.5 W. Both the dc optimizer-based system and the DPP-based system then use a string-level inverter to convert dc string voltage into ac voltage. In a microinverter system, the PV module voltage is directly converted to ac, but the efficiency of a microinverter is typically lower and its per-watt cost is typically higher than that of a string-level inverter or a central inverter. Table I summarizes the power losses of these three different solutions for the irradiance condition in this example. DC optimizers are assumed to have an average efficiency of 96%, while that of DPP converters is assumed to be 92% since they often operate in light-load conditions. For both the dc optimizer- and DPP-based system,

Manuscript received January 31, 2014; revised April 28, 2014; accepted May 28, 2014. Date of publication June 12, 2014; date of current version November 3, 2014. The information, data, or work presented herein was supported by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award DE-AR0000217. Recommended for publication by Associate Editor V. Agarwal.

The authors are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA (e-mail: sqin3@illinois.edu; scady2@illinois.edu; aledan@illinois.edu; pilawa@illinois.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPEL.2014.2330335

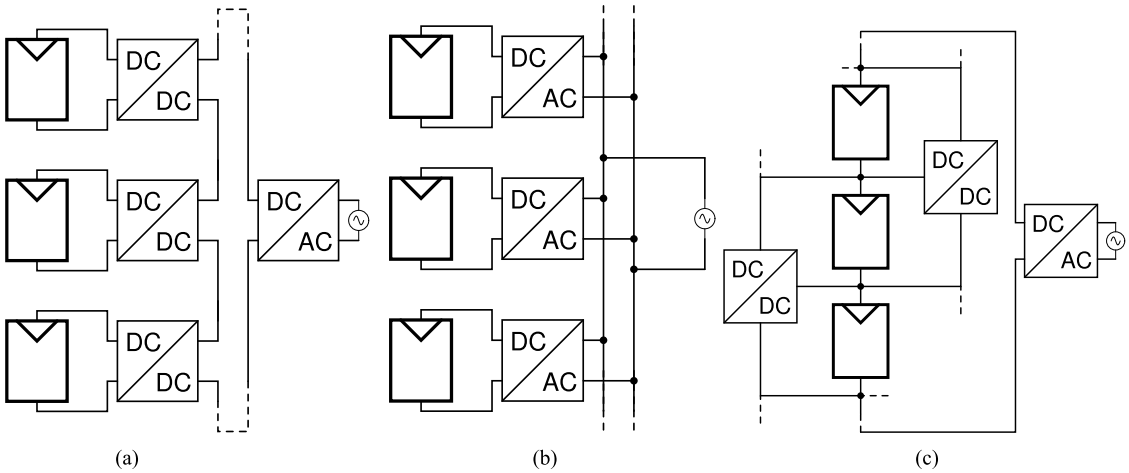


Fig. 1. Three types of distributed power electronics solutions for PV systems. (a) DC optimizer. (b) Microinverter. (c) DPP-based system.

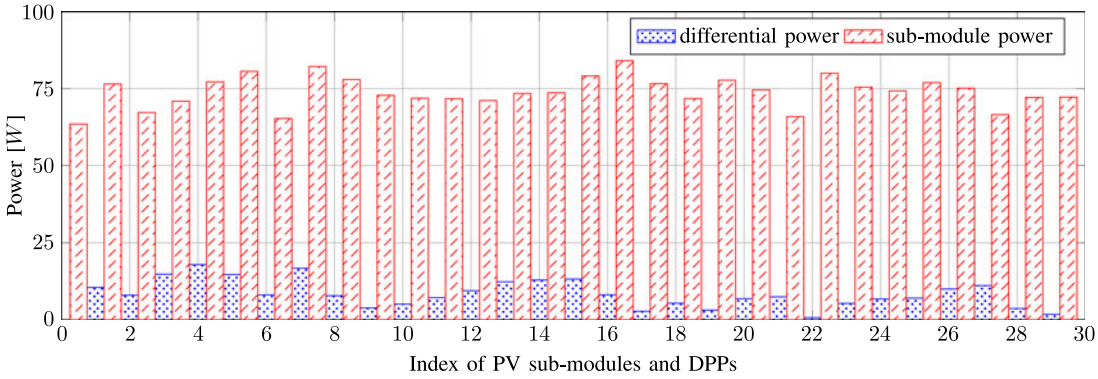


Fig. 2. Maximum power of PV submodules and differential power of a 30 PV submodule system with randomly generated irradiance condition.

TABLE I
COMPARISON OF SYSTEM EFFICIENCY OF THE DC OPTIMIZER-BASED SYSTEM, THE MICROINVERTERS AND THE DPP-BASED SYSTEM

Losses \ Solutions	DC optimizer	Microinverter	DPP
Power processed by dc stage [W]	2218	N/A	242.5
Intermediate power losses [W]	$2218 \times 4\% = 88.7$	N/A	$242.5 \times 8\% = 19.4$
Power processed by inverter(s) [W]	$2218 - 88.7 = 2129.3$	2218	$2218 - 19.4 = 2198.6$
Inverter power losses [W]	$2129.3 \times 2\% = 42.6$	$2218 \times 5.0\% = 110.9$	$2198.6 \times 2\% = 44.0$
Overall efficiency	94.08%	95.00%	97.14%

a string inverter efficiency of 98% was assumed. The microinverters are assumed to have an efficiency of 95%, and since, in practice, they only interface with PV modules, and therefore cannot compensate mismatch between submodules, the effective efficiency is typically even lower. As is evident from Table I, the DPP-based system yields significant advantages in terms of overall system efficiency.

To apply the concept of DPP to PV systems, especially at the submodule-level, several architectures and corresponding control schemes have been proposed [9]–[11], [13]–[17]. For example, the work presented in [14] and [17] uses a switched inductor topology; the work presented in [9] uses a resonant switched capacitor topology; the work presented in [10] uses a transformer coupled topology. All of these proposed solutions adopt a control method commonly known as voltage equalization or "virtual parallel" operation, in which all distributed converters strive to equalize the voltages of all PV submodules in a string. The distributed converters in the system run in open loop, simplifying the control requirements while obviating the need for communication among converters. However, the voltage equalization approach can only achieve *near* maximum power point (MPP) operation without truly seeking the MPP of each individual submodule. The effectiveness of this approach relies on the fact that the MPP voltages of submodules in a string are very close even if their MPP currents differ significantly. This does require all the submodules to have very

TABLE II
COMPARISON OF DPP CONTROL APPROACHES

Features \ Approaches	Virtual Parallel [9], [10], [14]	Centralized P&O [11], [15], [16]	Multilevel PPT [13]	This work
Tracking	Near MPPT	True MPPT	True MPPT	True MPPT
Distributed algorithm	Yes	No	Yes	Yes
Requiring local current sensing	No	Yes	Yes	No
Requiring communication	No	Centralized	Synchronization	Neighbor-to-Neighbor

similar electrical characteristics, which in practice is typically guaranteed by a costly binning process performed by PV module manufacturers [18]. However, even modules that are carefully matched at installation will suffer from nonuniform degradation after several years of field exposure. As illustrated in [19], the standard deviation of PV module MPP voltages may increase by nearly four times over a 20-year period. Furthermore, field operating conditions that result in thermal gradients or severe irradiance mismatch along the PV string can also cause the MPP voltages of submodules to drift apart. All of these factors limit the tracking efficiency of the voltage equalization approach; thus, in order to increase the total energy harvested over the entire lifetime of the PV system, it is necessary to develop a scheme capable of true MPPT.

On the other hand, the generation control circuit presented in [15] and [16] achieves true MPPT without any local current sensing at each submodule, but employs a control scheme that requires communication between all converters and a central control unit. The central control unit has to i) command all the DPP converters to exhaust every possible combination of converter duty ratio perturbations, and ii) measure the string voltage, respectively, in sequence before making the next DPP tracking step. For a system with n DPP converters, the central control unit has to try 2^n duty ratio perturbations during each DPP tracking step, rendering the algorithm slow, and infeasible for a large system. Moreover, with this approach, the reliability of the system is fundamentally limited, as a single failure of the central control unit, or a communication link failure, would result in the malfunction or complete loss of the overall system. The work presented in [11] adopts a buck–boost converter topology that overcomes some of the limitations in the architecture of the generation control circuit, and achieves true MPPT with a faster control scheme. However, this control scheme still requires communication between all DPP converters and the central converter. Recently, a distributed MPPT approach for a DPP-based system was presented in [13], which significantly reduces the communication requirements. However, this approach requires measurements of all PV submodule currents; this results in additional power losses. Moreover, this approach requires synchronization between power converters, which still relies on some communication.

Table II provides a comparison of some key features of DPP control approaches presented in previous work and the one proposed in this paper. The information in this table implies that for DPP converters, either communication or local current measurements are needed to acquire adequate information to perform true MPPT operation. Implementing local current measurement impairs the system efficiency and increases the hardware cost, whereas implementing communication may result in a much smaller impact because communication hardware is often required for other purposes, including individual PV module diagnostics and on/off capability. Therefore, our suggested approach is to eliminate local current sensing while preserving communication, but to reduce the communication requirement to overcome the limitations of the centralized approaches. In this paper, we present a distributed algorithm that requires only *neighbor-to-neighbor* communication between adjacent DPP converters to perform submodule-level *true* MPPT. In our approach, there is *no* need for a central control unit or any local current sensing. Each DPP only perturbs its duty ratio once per DPP tracking step, so for a system with n DPP converters, only n duty ratio perturbations are required per DPP tracking step (instead of 2^n perturbations in previous solutions [15]); each DPP converter observes only local voltage changes (instead of the entire string voltage). Moreover, the algorithm has the potential for parallelization of the duty ratio perturbations as the sensitivity between nonadjacent DPP converters is low, which can reduce the communication overhead and speed up the tracking even further. Compared to the centralized counterparts, no extra hardware is needed since the algorithm can be implemented in the DPP converter microcontrollers that already exist for converter local control. Furthermore, since a central control unit is unnecessary and each microcontroller can make independent control decisions, there is no longer a single point of failure, increasing the overall system reliability.

The remainder of this paper is organized as follows: the formulation of the proposed distributed MPPT algorithm is presented in Section II. Some properties of the proposed algorithm are discussed in Section III. Hardware implementation of the DPP converters and the experimental setup used to verify the effectiveness of the MPPT algorithm are introduced in Section IV. Section V provides case studies involving a three-DPP system, and a five-DPP system, including both simulation and experimental results, followed with simulation results for a larger 31-DPP system. Finally, Section VI concludes the paper.

II. MPPT ALGORITHM FORMULATION

In this section, we formulate a distributed iterative algorithm which, through the exchange of information among local controllers, and for a given string current, maximizes the power extracted from an array of n series-connected PV submodules outfitted with $n-1$ DPP converters. We first formulate the algorithm for a three-submodule, two-DPP system, and then generalize it to a system of any size.

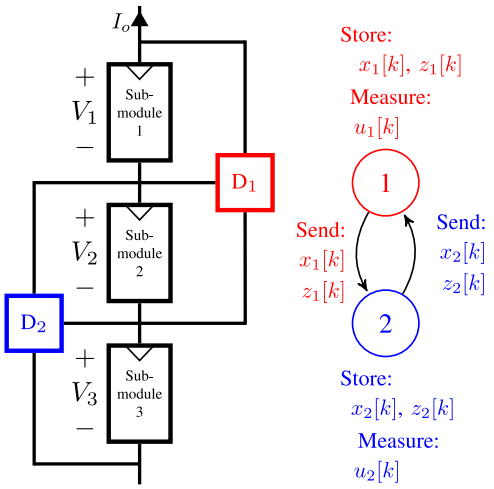


Fig. 3. Block diagram of a three-submodule, two-DPP system.

A. *Algorithm Formulation for a Three-Submodule, Two-DPP System*

A system comprising two-DPP converters and three-submodules is illustrated in Fig. 3. This system is based on the architecture presented in [11]. Each DPP is implemented as a bidirectional buck–boost converter that enforces a voltage ratio between two adjacent PV submodules. The PV string is attached to a central converter (omitted in Fig. 3). The control objective of the MPPT algorithm is to maximize the power extracted from the PV string, P_o, i.e.,

$$\underset{D_1,D_2}{\text{maximize}}\ P_o = V_o \times I_o, \tag{1}$$

where I_o is the string current, and $V_o := V_1 + V_2 + V_3$ is the string voltage. Given an irradiance condition, each PV submodule has a unique MPP $(V_{i,\max}, I_{i,\max})$, $i = 1, 2, 3$. As derived in [8], when each PV submodule is operating at its respective MPP, the entire system is operating at its global MPP, and the corresponding string voltage and string current can be uniquely determined through KCL and KVL analysis as $V_{o,\max}$ and $I_{o,\max}$, respectively. Moreover, the duty ratios of all DPP converters, $D_{1,\max}$ and $D_{2,\max}$, can be uniquely determined, such that $I_{o,\max}$, $D_{1,\max}$ and $D_{2,\max}$ form a set of variables that fully determines the system. The control objective, thus, entails tracking the unique combination of $(I_{o,\max}, D_{1,\max}, D_{2,\max})$ that corresponds to $P_{o,\max}$.

To this end, we separate the setting of I_o and V_o into two control loops. In a relatively slow control loop, the central converter is configured as a controllable current sink, and it performs a conventional perturb and observe (P&O) algorithm to seek the string current I_o that gives the largest P_o. In the second, relatively fast control loop, all DPP converters adjust their duty ratios in each DPP tracking iteration and take many iterations to maximize V_o. Since the perturbation interval of the central converter is much longer than the time constant of the DPP converter control loop (because the DPP converters are switching at a much higher frequency, as will be discussed in Section IV), at any given time, the string current, I_o, can be considered temporarily

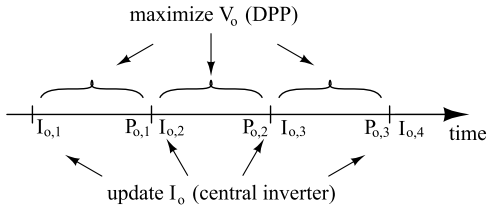


Fig. 4. Timeline of the "slow" control loop (central inverter) and "fast" control loop (DPP).

fixed for the DPPs. Given a fixed I_o, it is easy to see that, regardless of the irradiance incident on each of the submodules, maximizing the string voltage V_o is equivalent to maximizing P_o.

The timeline of Fig. 4 illustrates the operation of the two control loops: in the slow control loop, the central inverter employs a P&O algorithm, such that it first updates and enforces ("perturb") a certain string current value I_o. Following this, the DPP converters take many iterations in the fast control loop to maximize the string voltage V_o for this I_o. The string power is then measured ("observed") by the central inverter to use in its P&O algorithm. The slow and fast control loops can repeat this pattern and find the optimal string current $I_{o,\max}$. Here, we assume that the slow loop implemented in the central inverter uses a conventional P&O algorithm to seek $I_{o,\max}$, and we focus on the distributed control algorithm for DPPs in the fast control loop.

In our system architecture, we observe that each DPP has access to measurements of only two of the three submodule voltages instead of the entire string. Thus, in order to determine the duty ratios for which V_o is maximized, we assume that the local controllers of DPP_1 and DPP_2 can share information through neighbor-to-neighbor communication; this can be described by a directed graph, as shown on the right of the block diagram in Fig. 3. Then, we tailor the distributed optimization algorithm in [20] to our setting. [The reader is referred to the Appendix for a brief overview on the mathematics of this algorithm.]

The control objective described above can be written as

$$\underset{D_1,D_2}{\text{maximize}}\quad V_o(D_1, D_2) = V_1(D_1, D_2) + V_2(D_1, D_2) + V_3(D_1, D_2),$$

where D_1 and D_2 are the duty ratios of DPP_1 and DPP_2, respectively. Then, the maxima of the function $V_o(D_1, D_2)$ can be found by setting its gradient to zero, as in (2) at the bottom of next page, where D_1^* and D_2^* are the respective duty ratios of DPP_1 and DPP_2 that correspond to the MPP. To find D_1^* and D_2^*, the local controller of each DPP iteratively adjusts its duty ratio based on i) local voltage measurements, ii) state variables maintained locally, and iii) variables maintained by neighboring DPPs. Let $k = 1, 2, \ldots$ index the iterations performed by every DPP, and let $x_1[k]$, $x_2[k]$ be state vectors maintained by the local controllers of DPP_1 and DPP_2, respectively. For DPP_1, the entries of $x_1[k]$ are $D_1[k]$—the actual duty ratio of DPP_1, and $\hat{D}_{1,2}[k]$—DPP_1's estimate of the duty ratio of DPP_2. Similarly, for DPP_2, the entries of $x_2[k]$ are $\hat{D}_{2,1}[k]$—DPP_2's estimate

of the duty ratio of DPP$_1$, and $D_2[k]$—the actual duty ratio of DPP$_2$. Furthermore, let $z_1[k]$ and $z_2[k]$ be ancillary state vectors maintained by the local controllers, respectively; unlike $x_1[k]$ and $x_2[k]$, these ancillary states do not correspond to any physical variable. For the entire system, we define

$$x[k] = [\underbrace{D_1[k], \hat{D}_{1,2}[k]}_{x_1[k]}, \underbrace{\hat{D}_{2,1}[k], D_2[k]}_{x_2[k]}]^{\mathrm{T}}, \tag{3}$$

$$z[k] = [\underbrace{z_{1,1}[k], z_{1,2}[k]}_{z_1[k]}, \underbrace{z_{2,1}[k], z_{2,2}[k]}_{z_2[k]}]^{\mathrm{T}}. \tag{4}$$

Then, at each iteration, the variables are updated as

$$x[k+1] = (\mathbb{I}_4 - \delta\tilde{L})x[k] - \delta\tilde{L}z[k] + \delta\gamma u[k], \tag{5}$$

$$z[k+1] = z[k] + \delta\tilde{L}x[k], \tag{6}$$

where $\mathbb{I}_4$ is the 4×4 identity matrix; $\tilde{L} = L \otimes \mathbb{I}_2$, where $\mathbb{I}_2$ is the 2×2 identity matrix,

$$L = \begin{bmatrix} 1 & -1 \\ -1 & 0 \end{bmatrix}$$

is the Laplacian matrix of the graph representing the exchange of information between local controllers as depicted in Fig. 3, and "$\otimes$" denotes the Kronecker product of L and $\mathbb{I}_2$ (see, e.g., [21]); δ and γ are parameters that can be used to tune the algorithm (we will discuss this matter in detail in Section III); and $u[k]$ is defined as in (7) at the bottom of this page, with φ_1 and φ_2 defined, respectively, as,

$$\varphi_1(D_1, D_2) := V_1(D_1, D_2) + \frac{1}{2}V_2(D_1, D_2)$$

$$\varphi_2(D_1, D_2) := \frac{1}{2}V_2(D_1, D_2) + V_3(D_1, D_2).$$

From the aforesaid discussion, we see that the update of DPP$_1$'s states, $x_1[k+1]$ and $z_1[k+1]$, depends on its own states, $x_1[k]$ and $z_1[k]$, the states of its neighbor, $x_2[k]$ and $z_2[k]$, and the partial derivatives of the submodule voltages to which DPP$_1$ is directly attached to, i.e., $u_1[k]$. In this regard, while the necessary information that each DPP converter needs for updating (5) and (6) can be acquired through neighbor-to-neighbor communication, in order to obtain $u_i[k]$, each DPP$_i$ must estimate the partial derivatives of the $\varphi_i(\cdot)$ function. To perform this estimation, at every iteration, each DPP$_i$ alternatively perturbs its duty ratio by a fixed small amount ΔD_i, while both local controllers observe the submodule voltages; the timeline of this process is depicted in Fig. 5. After both local controllers have observed the results of each perturbation, and upon receiving the necessary information from the neighboring DPP, they can approximate the partial derivatives as $\frac{\partial\varphi_i}{\partial D_j} \approx \frac{\Delta\varphi_i}{\Delta D_j}$, $i, j = 1, 2$, and update their respective state variables. After $k = m$ iterations, for m sufficiently large, we have that $D_1[m] \approx D_1^*$ and $D_2[m] \approx D_2^*$; thus, the string voltage is maximized and the maximum power is extracted from the system at a given string current I_o. The fast control loop continuously tracks the maximum power and once the slow control loop updates I_o to a new value, the fast control loop will maximize the output power for this new condition. The flowchart in Fig. 6 summarizes how the two control loops operate and interact. Note that the two control loops are not time synchronized, but operate independently of one another.

B. Algorithm Formulation for an n-Submodule, (n−1)-DPP System

For a system of n submodules and $n-1$ DPP converters, the communication graph of which is shown in Fig. 7, we define the state vector, and the ancillary state vector of the entire system as

$$x[k] = \begin{bmatrix} x_1[k] & x_2[k] & \dots & x_i[k] & \dots & x_{n-1}[k] \end{bmatrix}^{\mathrm{T}}, \tag{8}$$

$$z[k] = \begin{bmatrix} z_1[k] & z_2[k] & \dots & z_i[k] & \dots & z_{n-1}[k] \end{bmatrix}^{\mathrm{T}}, \tag{9}$$

where $x_i[k]$ and $z_i[k]$, $i = 1, 2, \dots, n-1$, are state vectors maintained by the local controller of DPP$_i$. Each local controller maintains an estimate of the duty ratios of all other DPPs in addition to its own; thus, $x_i[k]$ consists of $\hat{D}_{i,j}[k]$, $i, j = 1, \dots, n-1$, $i \neq j$ and $D_i[k]$, i.e.,

$$x_i[k] = \begin{bmatrix} \hat{D}_{i,1}[k], \dots, D_i[k], \dots, \hat{D}_{i,n-1}[k] \end{bmatrix}^{\mathrm{T}},$$

$$z_i[k] = \begin{bmatrix} z_{i,1}[k], z_{i,2}[k], \dots, z_{i,n-1}[k] \end{bmatrix}^{\mathrm{T}}.$$

We also define the input vector as

$$u[k] = \begin{bmatrix} \dfrac{\partial\varphi_1(D)}{\partial D} & \dfrac{\partial\varphi_2(D)}{\partial D} & \dots & \dfrac{\partial\varphi_{n-1}(D)}{\partial D} \end{bmatrix}^{\mathrm{T}}, \tag{10}$$

$$\nabla V(D_1^*, D_2^*) = \begin{bmatrix} \left.\frac{\partial V_1(D_1,D_2)}{\partial D_1}\right|_{D_1^*,D_2^*} + \left.\frac{\partial V_2(D_1,D_2)}{\partial D_1}\right|_{D_1^*,D_2^*} + \left.\frac{\partial V_3(D_1,D_2)}{\partial D_1}\right|_{D_1^*,D_2^*} \\ \left.\frac{\partial V_1(D_1,D_2)}{\partial D_2}\right|_{D_1^*,D_2^*} + \left.\frac{\partial V_2(D_1,D_2)}{\partial D_2}\right|_{D_1^*,D_2^*} + \left.\frac{\partial V_3(D_1,D_2)}{\partial D_2}\right|_{D_1^*,D_2^*} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2}$$

$$u[k] := \underbrace{\left[\left.\frac{\partial\varphi_1}{\partial D_1}\right|_{D_1[k],D_2[k]}, \left.\frac{\partial\varphi_1}{\partial D_2}\right|_{D_1[k],D_2[k]}\right]}_{u_1[k]=\frac{\partial\varphi_1(D)}{\partial D}}, \underbrace{\left[\left.\frac{\partial\varphi_2}{\partial D_1}\right|_{D_1[k],D_2[k]}, \left.\frac{\partial\varphi_2}{\partial D_2}\right|_{D_1[k],D_2[k]}\right]}_{u_2[k]=\frac{\partial\varphi_2(D)}{\partial D}}^{\mathrm{T}} \tag{7}$$

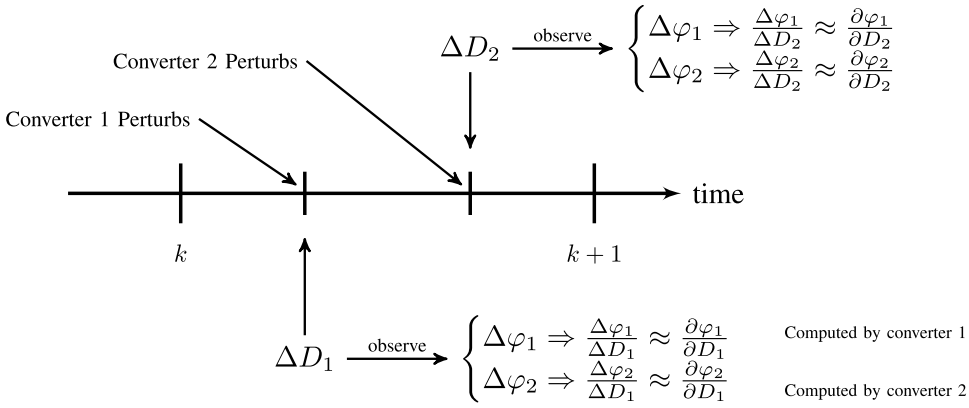


Fig. 5. Timeline of distributed P&O for the three-submodule, two-DPP system.

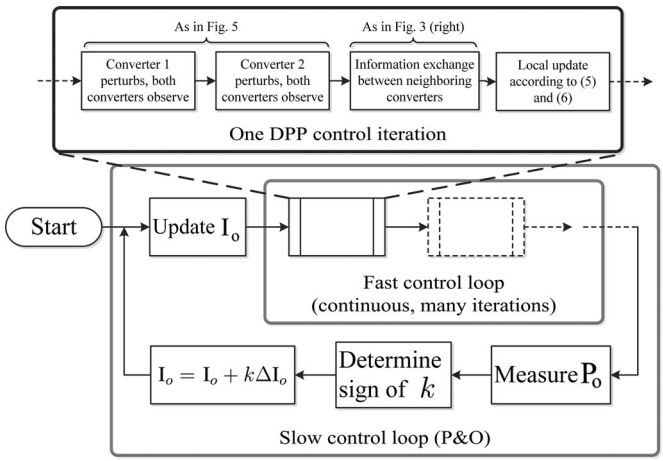


Fig. 6. Simplified flowchart of the proposed algorithm for the three-submodule, two-DPP system.

where

$$\frac{\partial\varphi_i(D)}{\partial D} = \left[\frac{\partial\varphi_i(D)}{\partial D_1}, \frac{\partial\varphi_i(D)}{\partial D_2}, \ldots, \frac{\partial\varphi_i(D)}{\partial D_{n-1}}\right]^{\mathrm{T}}, \quad (11)$$

with $\varphi_i(D)$ given by

$$\varphi_i(D) = \begin{cases} V_i(D) + \frac{1}{2}V_{i+1}(D), & i = 1, \\ \frac{1}{2}V_i(D) + \frac{1}{2}V_{i+1}(D), & 1 < i < n-1, \\ \frac{1}{2}V_i(D) + V_{i+1}(D), & i = n-1. \end{cases} \quad (12)$$

Then, at each iteration, similar to (5) and (6), the variables are updated as

$$x[k+1] = (\mathbb{I}_{(n-1)^2} - \delta\tilde{L})x[k] - \delta\tilde{L}z[k] + \delta\gamma u[k], \quad (13)$$

$$z[k+1] = z[k] + \delta\tilde{L}x[k], \quad (14)$$

where $\mathbb{I}_{(n-1)^2}$ is the $(n-1)\times(n-1)$ identity matrix; $\tilde{L} = L \otimes \mathbb{I}_{n-1}$, where L is now the Laplacian matrix of the graph in Fig. 7:

$$L = \begin{bmatrix} 1 & -1 & 0 & \ldots & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & 0 \\ 0 & -1 & 2 & -1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & \cdots & -1 & 1 \end{bmatrix}.$$

By inspecting (13) and (14) for the n-submodule case, it is easy to see that the state of DPP$_i$ at instant $k+1$, $x_i[k+1]$ and $z_i[k+1]$, depends on its own state at instant k, $x_i[k]$ and $z_i[k]$; the states of its neighboring DPPs at instant k, i.e., $x_{i-1}[k]$, $z_{i-1}[k]$, $x_{i+1}[k]$, $z_{i+1}[k]$, all of which can be acquired through the neighbor-to-neighbor communication; and $u_i[k]$, which is approximated by DPP$_i$ as $\frac{\Delta\varphi_i(D)}{\Delta D}$. Similar to the two-DPP case, after $k = m$ iterations, for m sufficiently large, we have $D_i[m] \approx D_i^*, i = 1, 2, \ldots, n-1$.

Note that although the algorithm update functions are compactly written in matrix form in (13) and (14), the actual data storage and computation is distributed among the DPPs. To understand how this algorithm is distributed, consider DPP$_i$ (node i) in Fig. 7 as an example. DPP$_i$ locally stores and updates vector $x_i[k]$ and $z_i[k]$ that are part of (8) and (9), respectively. For the computation performed by DPP$_i$, only the ith row of Laplacian matrix L (representing the information exchange of DPP$_i$ with its closest neighbors) is relevant. If we expand (13) and (14), and write down only the part needed to compute $x_i[k+1]$, we obtain that

$$\begin{aligned} x_i[k+1] = {} & \delta x_{i-1}[k] + (1-2\delta)x_i[k] + \delta x_{i+1}[k] \\ & + \delta z_{i-1}[k] - 2\delta z_i[k] + \delta z_{i+1}[k] + \delta\gamma u_i[k], \\ z_i[k+1] = {} & z_i[k] - \delta x_{i-1}[k] + 2\delta x_i[k] - \delta x_{i+1}[k], \end{aligned}$$

which illustrates that the computation of $x_i[k+1]$ can be performed independently by DPP$_i$ as long as this DPP converter has access to the state of its closest neighbors (this is achieved in

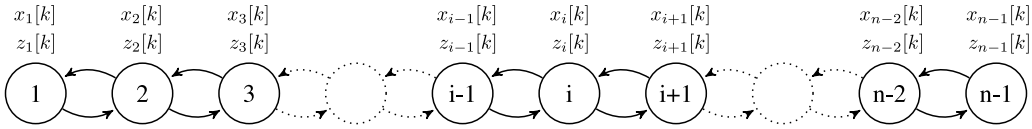


Fig. 7. Graph of an n-submodule, $(n-1)$-DPP system.

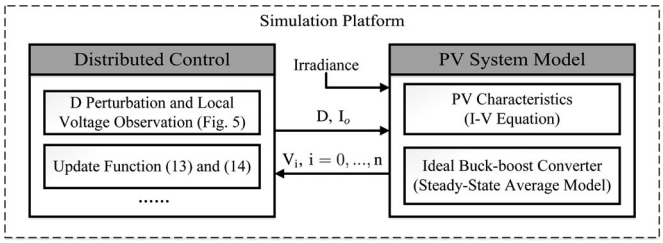


Fig. 8. Components of the simulation platform.

practice through neighbor-to-neighbor communications). Other states of the system are irrelevant to DPP_i's computation. A similar conclusion applies to other DPP converters; therefore, each DPP converter performs only part of the update computation relevant to itself, and thus the proposed algorithm is distributed.

III. Discussion on Algorithm Characteristics

In this section, we first briefly introduce the computer simulation platform of the PV system and the proposed algorithm. Following this, we consider a six-submodule, five-DPP system as an example, and illustrate via computer simulations the characteristics of the proposed algorithm in terms of i) update function parameter tuning, ii) communication topology, and iii) reconfiguration after communication failure.

A. Simulation Platform

We use MATLAB as the platform for all our numerical simulations. As shown in Fig. 8, the simulation platform consists of two independent parts: the PV system model block and the control block. The PV system model contains a set of equations that describes the electrical behaviors of the submodules and the DPP converters. The *I–V* characteristic of each PV submodule is described using the nonlinear equations based on the model presented in [22]. The buck–boost converters are modeled using the steady-state average equations [23]. Each converter takes a duty ratio command from the control block and enforces a corresponding voltage ratio between two neighboring PV submodules, and the Newton-Raphson method is applied to solve the set of equations of the PV system model. The control block of Fig. 8 runs the distributed algorithm as presented in Section II. During each control algorithm iteration, the control block passes duty ratio commands to the PV system block, which solves the system variables and returns the voltage information to the control block. The control block then uses the voltage information to calculate the update function, after which it starts a new iteration. Parameters in the control algorithm or the irradiance condition of PV submodules can be set before the simulation or changed during the simulation to study various situations.

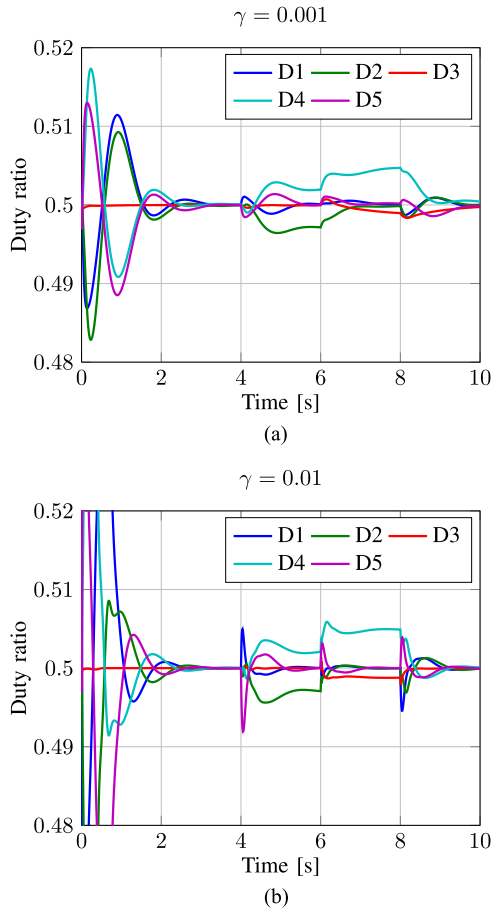


Fig. 9. Evolution of duty ratios computed by the distributed algorithm with fixed γ for a six-submodule, five-DPP system. (a) $\gamma = 0.001$. (b) $\gamma = 0.01$.

B. Choice of Tuning Parameter γ and δ

In (12), the parameter γ determines the influence that the input vector $u[k]$ has on the state vector $x[k]$. Typically, a relatively large value of γ renders a shorter response time but larger transient overshoot of duty ratios. As an example, Fig. 9(a) and (b) shows the duty ratio of a six-submodule, five-DPP system evolving over time with different values of γ in the update function; the normalized irradiance profile used in this example is as follows:

$$\begin{aligned}&\{100\%, 100\%, 100\%, 100\%, 100\%, 100\%\}\\ &\xrightarrow{t=4\text{ s}} \{100\%, 100\%, 80\%, 80\%, 50\%, 50\%\}\\ &\xrightarrow{t=6\text{ s}} \{100\%, 100\%, 100\%, 80\%, 50\%, 50\%\}\\ &\xrightarrow{t=8\text{ s}} \{100\%, 100\%, 100\%, 100\%, 100\%, 100\%\}, \quad (15)\end{aligned}$$

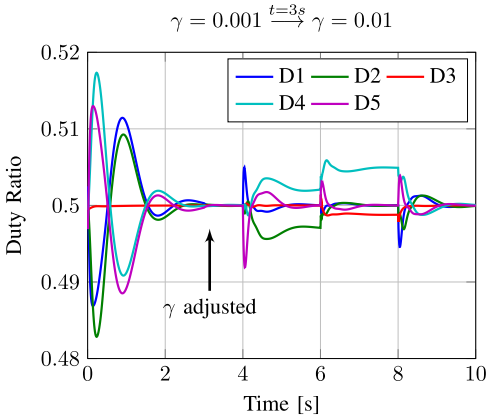


Fig. 10. Evolution of duty ratios computed by distributed algorithm with adaptive γ for a six-submodule, five-DPP system.

with step changes occurring at $t = 4, 6, 8$ s as indicated. As shown in Fig. 9(a) and (b), for different values of γ, the algorithm converges to the same steady-state value after each irradiance change. Compared to other transients, the duty ratio overshoot is very large at a "cold start," which corresponds to $t = 0$ s in Fig. 9(b). [At "cold start," the algorithm is just activated without receiving any information about the system.] Although the entries of $x[k]$ can be initialized to 0.5, the entries of $z[k]$ can only be initialized to 0. Since $z[k]$ is intended to balance the input vector $u[k]$ in (12), the state vector $x[k]$ is very sensitive to $u[k]$ when $z[k]$ is small; in this case, a smaller γ is preferable to suppress the overshoot. Note that the slow convergence resulting from this small γ is not a problem since, in the field, "cold start" conditions correspond to turning on the DPP system, which typically happens only once per day. Once the system converges and $z[k]$ reaches steady state, a small γ is undesirable due to its slow response, and a larger γ can be used for faster tracking upon irradiance changes (also referred to as "hot start"). Duty ratio overshoot in a "hot start" condition is typically small despite a large γ since the $z[k]$ has already reached a steady-state value. Therefore, some online tuning scheme for γ can be incorporated in the algorithm to improve its convergence properties; next, we discuss one such scheme.

To develop a method for smoothly changing the value of γ without introducing any disturbance to the duty ratio, we first note that after the algorithm converges to a steady-state value, we have that

$$x[k+1] = x[k] = x^*, \tag{16}$$

$$z[k+1] = z[k] = z^*. \tag{17}$$

Substituting (16) and (17) in (13) and (14), respectively, yields

$$\tilde{L}z^* = \gamma u^*, \tag{18}$$

$$\tilde{L}x^* = 0, \tag{19}$$

which suggests that scaling γ and z^* by the same multiplicative constant does not affect the equilibrium point of the algorithm. Therefore, the online tuning scheme can modify γ without introducing a disturbance on the duty ratio as long as $z[k]$ is simultaneously changed by the same scaling factor; Fig. 10 illustrates the response of a five-DPP system after such a tuning scheme is applied. The system "cold starts" with a small γ, and modifies the value of γ at $t = 3$ s without any noticeable disturbance to the duty ratios. The work in [24] presents the result of long-term measurement illustrating the transient effects of irradiance changes in solar PV applications, which can be used in the selection of γ for a suitable balance between speed and accuracy.

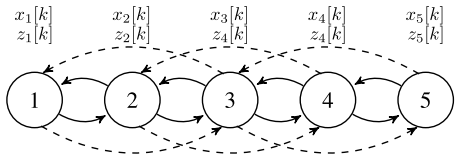


Fig. 11. Graph of a six-submodule, five-DPP system with redundant communications.

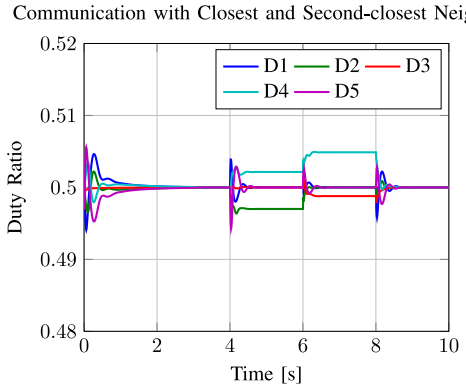


Fig. 12. Evolution of duty ratios computed by distributed algorithm with redundant communications.

Similar to the choice of γ, the choice of δ in (13) and (14) can also be increased to speed up the convergence of the algorithm. While γ controls the update iteration of $x[k]$ to $x[k+1]$ by scaling the influence of the gradient step as determined by $u[k]$, i.e., the larger the value of γ is, the larger the update of $x[k]$ to $x[k+1]$ will be, δ also scales the influence of the gradient step as determined by $u[k]$; but it also scales the influence of the auxiliary variable $\tilde{L}z[k]$; however, the signs of these terms are opposite, so it is not clear *a priori* which term will have more influence. On the other hand, δ is also the time elapsed between two consecutive iterations k and $k+1$; thus, by increasing δ, the iteration time is reduced proportionally. When choosing δ, there is a tradeoff between speed and stability of the algorithm. Because the system is highly nonlinear, δ can only be determined empirically, and we found δ should be kept smaller than 0.1 to guarantee stability in most scenarios.

C. Impact of Communication Topology on Convergence Speed

The convergence speed of the algorithm can be improved by modifying the communication topology; Fig. 11 illustrates a directed graph representing the communication topology of a six-submodule, five-DPP system. The same irradiance pattern

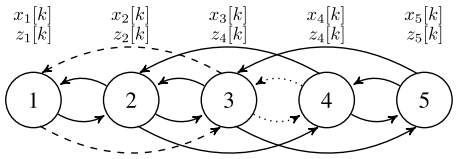


Fig. 13. Graph of a six-submodule, five-DPP system with communication failure.

in (15) is applied to the six submodules. By relying only on neighbor-to-neighbor communication (as captured by the solid line in Fig. 11), the evolution of duty ratios is relatively slow as has already been shown in Fig. 10. If redundancy is introduced to the communication topology, e.g., each DPP can communicate with its second closest neighbor(s) in addition to its direct neighbor(s) (as captured by the dashed line in Fig. 11), the convergence speed can be accelerated (as in Fig. 12) and the proposed algorithm can be easily adapted to accommodate this change. We only need to modify the Laplacian matrix, L, in the update functions in (13) and (14) to reflect the new communication topology, which in this case is given by

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ -1 & -1 & 4 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}. \tag{20}$$

Besides L, no further modification to the algorithm is necessary. As illustrated in Fig. 12, the extra communication links significantly improve convergence speed. Further simulation indicates that as we add more communication links, the convergence speed of the algorithm further increases. Although it seems that additional communication hardware is required in the real system to introduce the modification as discussed earlier, this is not really the case. As will be introduced in Section IV, our hardware design, which integrates three DPPs on one printed circuit board (PCB), readily lends itself to the communication topology in which each DPP can communicate with at least its first, second, and third neighbors in both directions without extra hardware.

D. Impact of Communication Topology on Reliability

As shown in [20], the proposed distributed algorithm converges to the optimal solution as long as the directed graph describing the communication topology remains strongly connected. Therefore, when the communication topology has redundancy, the algorithm can be reconfigured to work under communication link failures. Consider the six-submodule, five-DPP system again with the communication topology shown in Fig. 13. Each DPP is communicating with its first and second closest neighbors. The Laplacian matrix of this graph is given in (20). If a communication link fails as illustrated by the dashed lines in Fig. 13, and this failure is detected by the local controllers of the DPPs on each end of the link, the two controllers

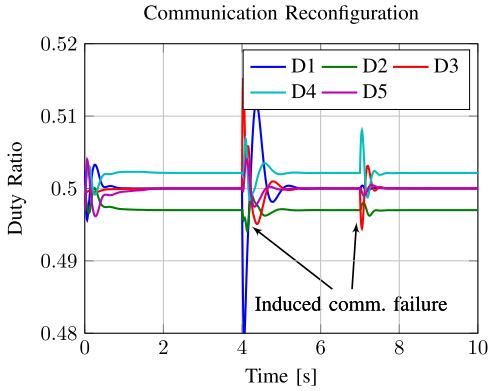


Fig. 14. Evolution of duty ratios computed by distributed algorithm with communication failure reconfiguration.

involved can simply reconfigure the algorithm by modifying the corresponding row of the Laplacian matrix used in their update function to reflect this change in the communication topology. Thus, the new Laplacian matrix is given by

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 3 & -1 & -1 \\ 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix} \tag{21}$$

note that only the first and third row in the Laplacian matrix change. This change is completely local to the two DPPs on the ends of the failed communication link, i.e., DPP_1 and DPP_3, and affects only the update functions of these two DPPs. The update functions for other DPPs remain the same and continue to perform MPPT, since other rows of the Laplacian matrix do not change. If another communication failure occurs after the first one, e.g., as illustrated by the dotted line in Fig. 13, the Laplacian matrix can be modified again; this results in

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 \\ 0 & -1 & 2 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ 0 & 0 & -1 & -1 & 2 \end{bmatrix}. \tag{22}$$

Fig. 14 illustrates the simulation result of this reconfiguration assuming unchanged irradiance pattern and the aforementioned communication link failures happening at $t = 4$ s and $t = 7$ s, respectively. As the simulation shows, after each communication failure, our algorithm is able to restore the duty ratio back to the optimal value after a short transient.

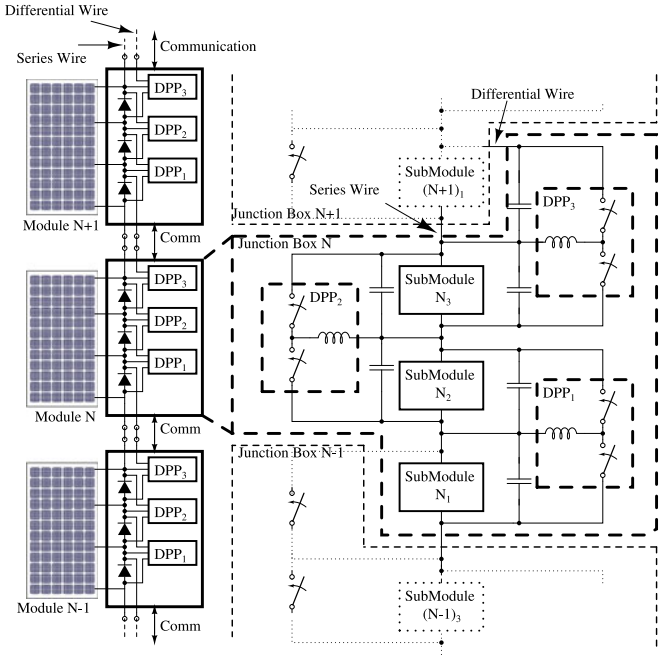


Fig. 15. PV junction box connection for a DPP system.

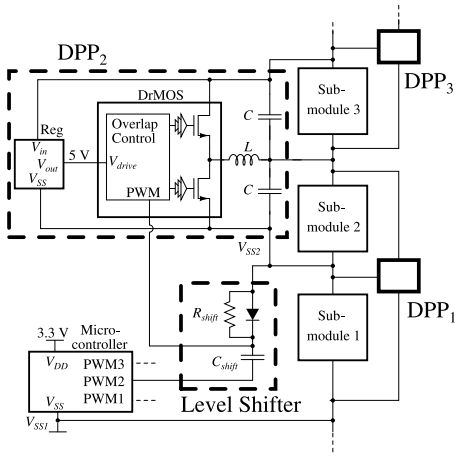


Fig. 16. Schematic of hardware design (details shown for DPP_2 only).

TABLE III
MAIN COMPONENT LIST

Device	Model	Value	Manufacturer
Microcontroller	STM32F105		STMicroelectronics
DRMOS	SiC780		Vishay
Linear Regulator	UA78L05CPK		Texas Instrument
L	SER1360-103KL	10 μH	Coilcraft
C	TMK212BBJ106KG-T	10 μF×4	Taiyo Yuden
$D_{\rm shift}$	1SS416CT(TPL3)		Toshiba
$C_{\rm shift}$	0402	4700 pF	
$R_{\rm shift}$	0402	100 kΩ	
I^2C Isolator	ADuM1250		Analog Devices

IV. LABORATORY TESTBED

In this section, a brief introduction to the DPP hardware design is provided, followed by a description of the experimental setup used to verify the effectiveness of the proposed algorithm.

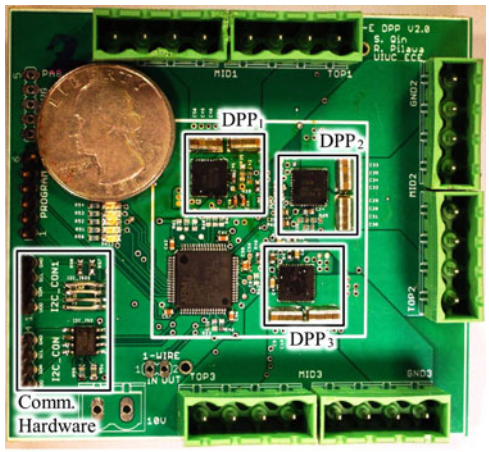
A. DPP Hardware Implementation

One of the primary goals of distributed power electronics for PV systems, including DPP-based systems, is to achieve PV module integration [25]. Nowadays, the cost of separate enclosures for distributed power electronics represents a very significant portion of the total system cost, motivating efforts to reduce the converter footprint to fit into the existing weather-resistant junction box of an off-the-shelf PV module. Therefore, for submodule DPP systems, the goal of the hardware design in this study is to achieve miniaturization of the DPPs for junction box integration, while maintaining high efficiency and capability for the implementation of the proposed distributed MPPT algorithm.

Fig. 15 illustrates the wire connection of a DPP system; typically, one PV module consists of three submodules. Therefore, each junction box needs to integrate three DPPs. Between adjacent junction boxes there are three wire connections: one series string connection in which the bulk power common to all submodules flows, one differential connection through which the DPPs shuffle the small power mismatch, and one or more wires for communication. We note that the bulk and differential power wires can be used for communication purpose, but that approach is not used in this study.

With the design objective described earlier in mind, we implement DPPs as bidirectional synchronous buck–boost converters. As analyzed in [11], the buck–boost topology allows for the use of low-voltage, fast-switching transistors. By employing a high switching frequency on the order of hundreds of kilohertz, the size of the magnetic components, which typically dominate the converter size, can be significantly reduced. Furthermore, while in [11] each DPP has one dedicated microcontroller, each microcontroller in our proposed implementation has enough peripherals to control more than one DPP. Moreover, since the three DPPs in one junction box are physically very close, they can be integrated on one PCB and controlled by one microcontroller. This eliminates the need for communication between DPPs in one junction box and reduces the control overhead in terms of hardware cost and power consumption. In this study, we chose a low-cost 32-bit ARM Cortex-M3 microcontroller, which also allows some flexibility for future expansions, e.g., implementation of fault-detection algorithms. Each DPP is constructed with a shielded power inductor and an integrated DRMOS power stage to achieve high efficiency and a very small footprint on the PCB; Table III contains a list of the main components used.

One unique challenge of controlling three DPPs with one microcontroller is sending the control signal across different voltage levels, as shown in Fig. 16. This figure illustrates the circuit details for DPP_2 only, while those for DPP_1 and DPP_3, which are similar, are omitted. In Fig. 16, while DDP_1 can be controlled directly by the microcontroller, the pulse width modulation (PWM) control signals sent to DPP_2 and DPP_3 must be level shifted because they do not have the same voltage



(a)

(b)

Fig. 17. Annotated photograph of the hardware prototype. (a) Front side of the test prototype. (b) Prototype fitting in a junction box.

reference as the microcontroller. The level-shifting circuitry employed in this study is implemented using low-cost passive devices (R_{shift}, C_{shift}, and D_{shift} in Fig. 16). When the microcontroller PWM_2 signal is low, the level-shifting capacitor gets charged to the voltage difference between the ground potential of DDP_2 and the ground potential of the microcontroller through the diode. The PWM input pin of DPP_2's DRMOS is, thus, pulled to DPP_2's ground potential. When the microcontroller PWM_2 signal reaches the high-voltage limit (3.3 V), the voltage of the DRMOS's PWM input pin is pushed to its high-voltage limit (3.3 V) with respect to DPP_2's ground reference, and the diode turns OFF. Since the PWM input pin of the DRMOS has a high impedance, any current that flows through the level-shifting capacitor is very small, so the input voltage to the DRMOS's PWM input pin can remain high for a long enough time before PWM_2 goes low again. A resistor is placed in parallel with the diode to prevent overvoltage across the PWM input pin due to a sudden decrease of submodule 1 voltage, i.e., the voltage difference between DPP_2's ground reference and the microcontroller's ground reference. The designer should be aware of certain considerations when selecting the value of level-shifting capacitors and resistors; factors to

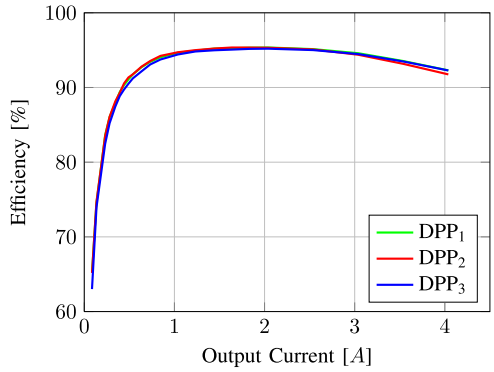


Fig. 18. Measured efficiency of the three DPPs on one hardware prototype at 10-V input voltage, 50% duty ratio.

consider include the time constant of the level shifter compared to the PWM frequency, and the driving capability of microcontroller PWM pins. For protection of the microcontroller PWM pin, a small value resistor can be placed in series with the level-shifting capacitor to reduce current spike during start up.

Fig. 17(a) shows an annotated photograph of the front side of the hardware prototype. Magnetic inductors are on the back side of the PCB. Note that a large portion of the board area is consumed by large connectors and ancillary circuitry to facilitate development and diagnosis, which can be eliminated in a final product. As shown in Fig. 17(a), all essential components of the hardware, including the inductors on the back side of the PCB, only take up a 3.75 cm$\times$3.75 cm area encompassed by a white rectangle, and, as shown in Fig. 17(b), can easily fit in a junction box.

A potential limitation of the digital PWM signal generated by the microcontroller is that with limited clock frequency, a tradeoff has to be made between PWM frequency and PWM resolution. The PWM frequency is determined by the requirement of high efficiency and small converter size, so it has to be maintained at a relatively high value (100 kHz in this study), rendering a low-PWM resolution given the limited microcontroller frequency. In each iteration of the MPPT algorithm proposed in Section II, the duty ratio updates that are calculated with high precision must be quantized to within the PWM resolution, impairing the effectiveness of the algorithm. To overcome this limitation, a PWM dithering technique was adapted from [26]–[28]. By dithering periodically between two adjacent quantized duty ratio values available from the microcontroller, with proper filtering, the effective PWM resolution in the hardware prototype is increased by ten times. The effective PWM resolution and other important specifications of the hardware prototype are listed in Table IV. Note that the DPP converter peak efficiency is above 95%. An efficiency characterization of the DPP converter across the entire load range is shown in Fig. 18. Note that the relatively low light-load efficiency can be greatly improved using pulse frequency modulation techniques [11].

B. *Experimental Setup*

To verify the proposed MPPT algorithm, we developed the indoor experimental setup shown in Fig. 19. A power supply (HP 6631A) was connected in parallel with each PV submodule

TABLE IV
HARDWARE PROTOTYPE SPECIFICATIONS

Submodule Voltage Range	3–13.4 V
Converter Power Rating	60 W
Switching Frequency	100 kHz
Duty Ratio Resolution (with PWM dithering)	1/3600
Converter Peak Efficiency	95%
Weight	28 g
Volume	8.575 cm^3

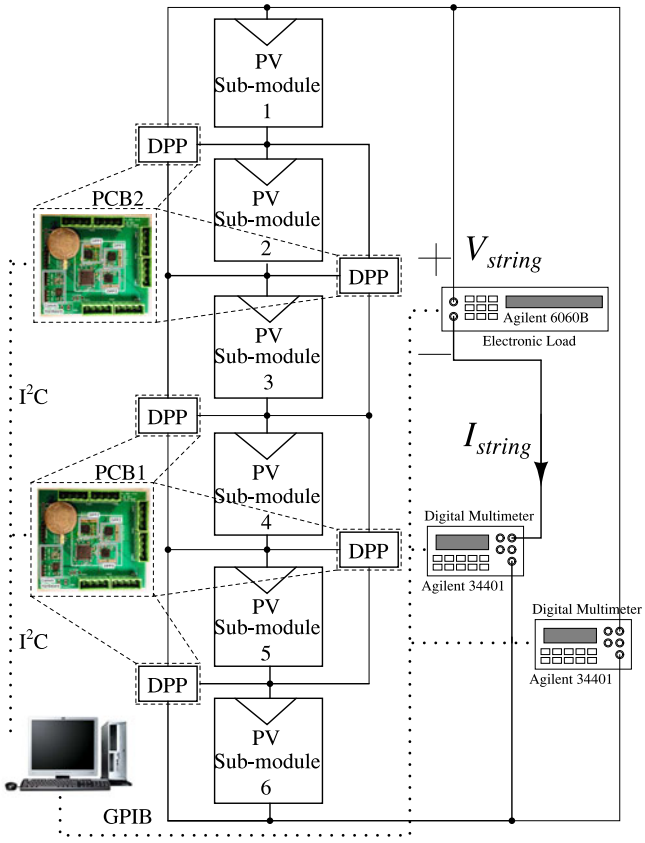


Fig. 19. Experimental setup.

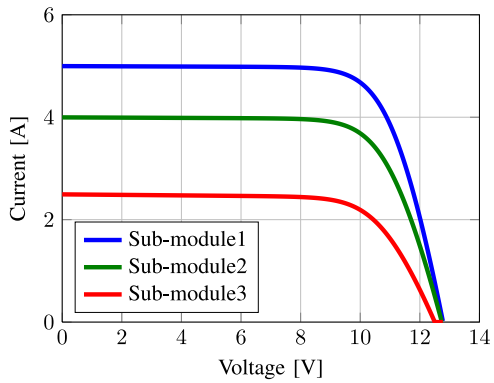


Fig. 20. *I–V* characteristics of three submodules with 100%, 80%, and 50% normalized irradiance.

to replicate the photo-generated current, providing an output *I–V* characteristic similar to the one that would result if the PV submodule were to be illuminated by sunlight. Details about this method to conduct repeatable indoor PV experiment can be found in [29]. Five DPPs implemented on two prototype boards were connected in parallel with a string of six PV submodules. Three of them are from a Sunmodule SW 235 Poly PV module, and the other three are from a SolarWorld Sunmodule SW 245 Poly PV module. These two modules are manufactured in the same process, but binned into two different models by the manufacturer for better matching of characteristics. In this experiment, they are chosen intentionally to illustrate that PV modules that undergo a less stringent binning process and, thus, have more mismatch can benefit from our proposed solution. An electronic load (Agilent 6060B) acted as a central converter and controlled the string current. Two digital multimeters (Agilent 34401) were used to measure the string voltage and current. All power supplies, digital multimeters, and the electronic load communicated with a computer through a GPIB interface. Microcontrollers on the two prototype boards communicated with each other and with the same computer through an I^2C interface. Note that for the following tests of the proposed distributed algorithm, the computer only received information from the microcontrollers for data logging and diagnostic purposes. The distributed algorithm, including data exchange and synchronization at each iteration, is fully implemented in the microcontrollers.

In the experiments, the current limits of the power supplies were set to certain percentages of the submodules' nominal short-circuit current to emulate the corresponding percentage of irradiance (e.g., a combination of 5 A, 4 A, and 2.5 A was used to emulate an irradiance profile of 100%, 80% and 50% normalized value). For a given irradiance mismatch scenario, the string current was kept fixed at a certain value by the electronic load to emulate the behavior of a central inverter during its P&O interval. The evolution of duty ratios of all DPPs, along with the string current and voltage after the convergence of the algorithm, was recorded by the computer.

V. CASE STUDIES

In this section, we verify the effectiveness of the proposed distributed algorithm through three case studies. We first present and compare simulation and experimental results from the simplest two-DPP system to illustrate its basic features. Then, we focus on the experimental result of a six-submodule five-DPP system and analyze its tracking efficiency. Finally, we demonstrate the scalability of the algorithm by presenting simulation results for a system containing 31 DPPs and 32 submodules.

A. Two-DPP System

To study the simplest two-DPP case, we conducted experiments on the setup discussed in Section IV but with only three submodules and compared the data with a MATLAB simulation. We set the irradiance pattern, string current, and initial duty ratios to be the same in the experiment and simulation,

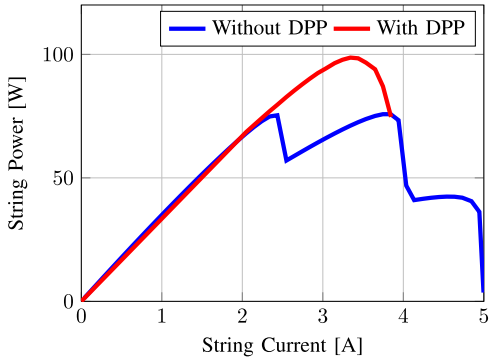


Fig. 21. String *P–I* characteristic of a two-DPP system with irradiance pattern of 100%, 80%, and 50%.

and recorded the evolution of the states of the algorithm. This test was performed for different conditions, and we found good matching between the simulation and the experimental data. For example, we set up a condition with $100\%, 80\%, \text{and } 50\%$ normalized irradiance on submodules 1, 2, and 3, respectively (Fig. 20 shows the measured corresponding *I–V* curves); We programmed the electronic load in Fig. 19 to initially perform a reference string current sweep, where the DPP converters were not operating. Following this, we performed a string current sweep with the DPP converters operating, executing the proposed algorithm. The DPP converters were allowed time for 200 iterations, which is enough time for the algorithm to converge to maximize the string voltage for that particular string current. The result of the string current sweeps is shown in Fig. 21. The DPP converters running the proposed algorithm smoothed out the *P–I* characteristics of the mismatched PV string and eliminated the local maxima. Therefore, a central inverter running the current referenced P&O method can converge to the optimal current given the concave *P–I* characteristics of the PV string with DPP converters. To illustrate the operation of the DPP control algorithm, we consider one situation in the current sweep as an example. When the string current is 3.3 A (corresponding to the global maximum power), Fig. 22(a) shows the evolution of the simulated duty ratios for this condition; Fig. 22(b) shows the evolution of the duty ratios computed using the experimental setup with the same condition. As the figures show, the state variables in both simulations and experiments quickly converge and roughly follow the same trajectory (also note that the duty ratios $D_1[k]$ and $D_2[k]$ settle to the same value as corresponding duty ratio estimates $\hat{D}_1[k]$ and $\hat{D}_2[k]$). To verify that the algorithm converges to the correct value, on the experimental setup, we also executed the centralized MPPT algorithm in [11], and recorded its duty ratio evolution as a benchmark (i.e., D_{ref1} and D_{ref2} in Fig. 22(b)). The final duty ratios of the proposed algorithm after convergence are almost identical to those of the centralized MPPT algorithm. The string voltage was measured as the proposed distributed algorithm converged, as shown in Fig. 22(c), which confirms the increase in string voltage. Tests of other irradiance patterns and string current exhibited similar results.

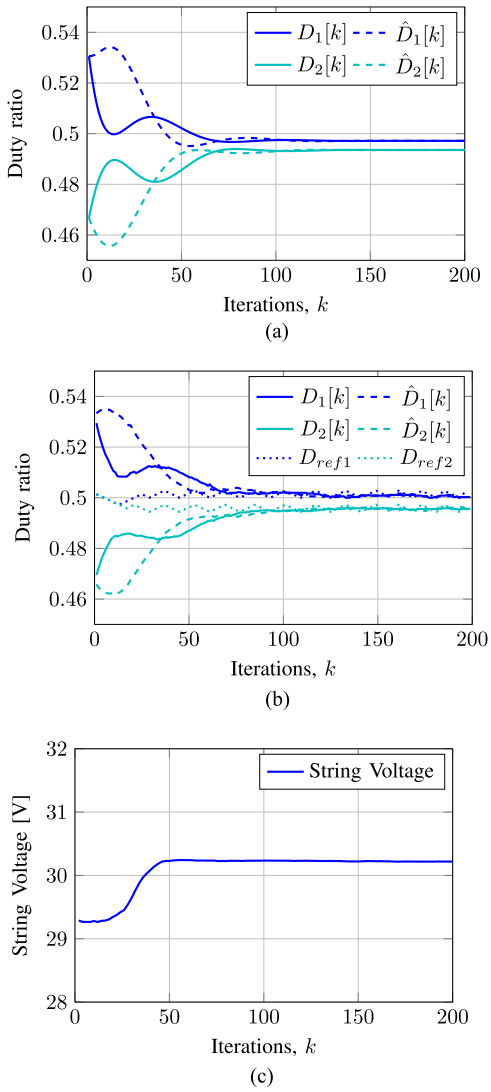


Fig. 22. Evolution of duty ratios computed by distributed algorithm for a two-DPP system with irradiance pattern of $100\%, 80\%, 50\%, I_{\text{string}} = 3.3\ A$, $\delta = 0.1$, and $\gamma = 0.003$. Experimental result of the evolution of the centralized algorithm proposed in [11] is also plotted in dotted lines in as a benchmark reference for the final value after convergence. The string voltage change in the experiment as a result of the evolution of the proposed distributed algorithm is also plotted. (a) Duty ratio simulation result. (b) Duty ratio experimental result. (c) String voltage experimental result.

B. *Five-DPP System*

Similar to the two-DPP case, we tested the proposed algorithm on a five-DPP system for different conditions. String current sweeps were performed in the same way as described for the two-DPP system. The *P–V* characteristics are similar to that of the two-DPP system and are omitted here. Again, we take one string current value in the current sweep steps as an example. Fig. 23(a) and (b) shows the simulation and experimental result for the same conditions and initial duty ratios. Note that among all the state variables, only the actual duty ratios $D_i[k], i = 1, 2, \ldots, 5$, are plotted (the corresponding $\hat{D}_i[k]$s are omitted). Along with the distributed MPPT algorithm, the centralized MPPT algorithm proposed in [11] was also executed,

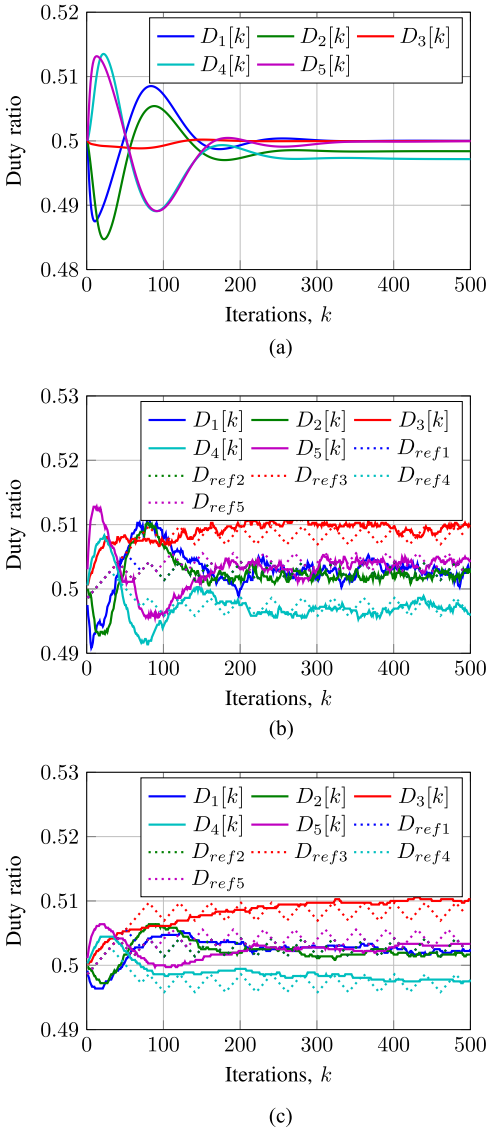


Fig. 23. Evolution of duty ratios computed by distributed algorithm for a five-DPP system with irradiance pattern of 100%, 100%, 80%, 80%, 50%, 50%, $I_{\text{string}} = 3.3$ A, $\delta = 0.1$, and different γ. Experimental result of the evolution of the centralized algorithm proposed in [11] is also plotted in dotted lines as a benchmark reference for the final value after convergence. (a) Simulation result, $\gamma = 0.002$. (b) Experimental result, $\gamma = 0.002$. (c) Experimental result, $\gamma = 0.0004$.

and its evolution is plotted in Fig. 23(b) as a benchmark. As the algorithm converged, an increase in string voltage was experimentally measured with similar characteristics as the two-DPP case. Note that in both the two-DPP and the five-DPP systems, states in simulation and experiment follow very similar trajectories but do not converge to exactly the same final values; this is primarily due to two reasons. First, the simulation does not take into account the converter power losses. Second, as we used two types of PV modules in the experiment to emulate modules with less stringent binning, there is some variation in the *I–V* curves of the six submodules, which is not captured in the simulation.

At the same time, it is evident that the proposed algorithm and the centralized MPPT algorithm converge to the same final

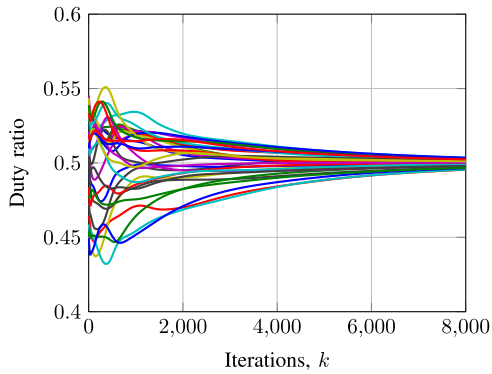


Fig. 24. Evolution of duty ratios computed by distributed algorithm for 31-DPP simulation with 100% irradiance on all submodules and random initial duty ratios.

value. However, due to the noise in the voltage measurements, the duty ratio values still have small variations after convergence. With a smaller value of γ in (13), the variation around the steady-state value can be significantly reduced as $D_i[k]$ is less dependent on $u_i[k]$, but the algorithm will take more iterations to converge to the steady-state value, as shown in Fig. 23(c), which is consistent with the analysis in Section III-B. In terms of hardware-related issues, the noise in the voltage measurements can be reduced by averaging multiple reading, or filtering with larger capacitors at the ADC pins, which may increase the time needed for each iteration. Therefore, a tradeoff between the convergence speed of the algorithm and the "noise" in the steady-state values should be carefully considered in practice.

To further verify that the proposed algorithm effectively tracked the MPP, we measured the static tracking efficiency, defined as the ratio of the actual power extracted to the maximum power of each submodule for a given irradiance, when the string current is at the optimal value. To acquire the actual power extracted from each submodule, we used two digital multimeters (Agilent 34401) to perform synchronized, fast (0.006 NPLC) measurements on the current and voltage of each submodule after the algorithm converged (the algorithm was kept running), so the tracking efficiency losses due to duty ratio perturbations, switching ripples, and other transients could be taken into account in these measurements. The measurements were performed 100 times continuously. Submodule power was calculated from each pair of current and voltage measurements, and averaged over the 100 times. On the other hand, to acquire the maximum submodule power, an output voltage sweep was performed on each submodule under the same irradiance, and the maximum output power was recorded. The resulting tracking efficiency was calculated as in in Table V for a few cases. Note that as introduced in Section IV-B, the six submodules used in the experiment were from two PV modules. Due to the internal wiring of the PV modules (three submodules connected in series in the junction box), the current of the submodule in the middle of each PV module was inaccessible, i.e., the current of submodule 2 and submodule 4 could not be directly measured when the algorithm was running. Therefore, we only measured the power extracted from the other submodules.

TABLE V
DUTY RATIO COMPARISON AND TRACKING EFFICIENCY AFTER ALGORITHM CONVERGENCE

Irradiance Pattern	100%, 100%, 80%, 80%, 50%, 50%	100%, 100%, 100%, 100%, 100%, 100%
Centralized MPPT Duty Ratio	0.503, 0.503, 0.508, 0.497, 0.503	0.499, 0.500, 0.502, 0.496, 0.501
Distributed MPPT Duty Ratio	0.502, 0.502, 0.510, 0.497, 0.504	0.499, 0.499, 0.502, 0.496, 0.501
Actual Submodule Power [W]	45.42, N/A, 35.33, 36.59, N/A, 22.34	45.16, N/A, 44.96, 45.99, N/A, 46.54
Maximum Submodule Power [W]	45.89, 45.63, 35.56, 36.67, 21.43, 22.35	45.89, 45.63, 45.39, 46.06, 45.00, 46.71
Tracking Efficiency	98.98%, N/A, 99.35%, 99.78%, N/A, 99.95%	98.41%, N/A, 99.05%, 99.85%, N/A, 99.63%

Another very important figure of merit is the dynamic MPPT efficiency of the system. It should be noted that the focus of this study is the control of DPP converters so that they can work with an *existing* central inverter design and compensate the mismatch between submodules in a PV array during partial shading, etc. While the DPP converters will improve the dynamic MPPT efficiency of the entire system by compensating for submodules mismatch, the dynamic efficiency of the system is determined and lower bounded by the central inverter, which is not the focus of this study. The dynamic MPPT efficiency will be explored in future field tests where an existing central inverter will be used.

C. 31-DPP System

To demonstrate the scalability of the proposed distributed MPPT algorithm, we extended the simulation to model a system with 32 submodules and 31 DPPs. The evolution of the duty ratios found by each of the DPP local controllers using the distributed algorithm is shown in Fig. 24, for randomly chosen initial conditions, and for an irradiance pattern in which all submodules receive 100% irradiance. As Fig. 24 shows the algorithm converges to the expected solution wherein the duty ratios are $D_i = 0.5$, $i = 1, \ldots, n-1$. While more iterations are required for a larger system to converge from a random initial point, some of the terms in $u_i[k] = \frac{\partial \varphi_i(D)}{\partial D_j} \approx \frac{\Delta \varphi_i(D)}{\Delta D_j}$, as defined in (11), are effectively zero when i and j are far apart. This allows parallelization of DPP duty ratio perturbations to speed up the convergence; we plan to explore this idea in future work.

VI. CONCLUSION

In this paper, we addressed the problem of maximizing the power extracted from a system of series-connected PV modules outfitted with DPPs. To tackle some of the drawbacks of previously-proposed centralized MPP algorithms, we introduced a distributed P&O algorithm that relies on neighbor-to-neighbor communication, and requires only local voltage measurements. Hardware suitable for PV junction box integration was developed. To verify the efficacy of the proposed algorithm, we presented simulation and experimental results for a two-DPP and a five-DPP system. We then demonstrated the scalability of the algorithm by presenting simulation results for a 31-DPP system.

APPENDIX

In this appendix, a brief introduction to the mathematical background of our proposed algorithm is presented; the reader is referred to [20] for a more detailed account. While [20] derives the algorithm in the continuous-time framework, the algorithm proposed in this study presents the discrete-time counterpart.

To derive the algorithm, we will start in the continuous-time framework and discretize the resulting dynamics afterwards. Consider a network with n nodes, i.e., $v_1, v_2, \ldots, v_n$, whose communication topology can be described by a strongly connected digraph $\mathcal{G}$, and each node v_i, $i \in \{1, \ldots, n\}$ has a continuously differentiable and convex function f^i that is only available to this node, then as illustrated in [20], the unconstrained optimization problem

$$\min f(x^o) = \sum_{i=1}^{n} f^i\left(x^o\right) \tag{23}$$

can be equivalently formulated as a constrained optimization problem

$$\min \tilde{f}(x) = \sum_{i=1}^{n} f^i\left(x_i\right), \tag{24}$$

$$\text{subject to } \tilde{L}x = 0_n, \tag{25}$$

where $\tilde{L} = L \otimes \mathbb{I}_n$; L is the Laplacian Matrix representing $\mathcal{G}$, $\mathbb{I}_n$ is the $n \times n$ identity matrix, and "$\otimes$" represents Kronecker product of L and $\mathbb{I}_n$; 0_n is the *n*-dimensional all-zero vector; x^i is node v_i's estimate of the solution to (23), and x is a vector containing estimates from all nodes, i.e.,

$$x = \left[x_1^T, x_2^T, \ldots, x_n^T\right]^T. \tag{26}$$

Then, this constrained optimization problem can be solved using augmented Lagrangian method, i.e.,

$$\min F(x, z) = \tilde{f}(x) + z^T \tilde{L}x + \frac{1}{2}x^T \tilde{L}x, \tag{27}$$

where z^T is the estimate of the Lagrange multiplier. To solve (27), we need to find the point (x^*, z^*) that sets the gradient of $F(x, z)$ to zero, which we can obtain through the following dynamics:

$$\dot{x} + \tilde{L}x + \tilde{L}z = -\nabla \tilde{f}(x), \tag{28}$$

$$\dot{z} = \tilde{L}x. \tag{29}$$

As (28) and (29) converge to the equilibrium point (x^*,z^*), x^* is the solution to (24), and $x_1 = x_2 = \cdots = x_n$ in (26) is the solution to the original problem in (23).

In our specific DPP system setting, the control objective is

$$\underset{D_1, \ldots, D_n}{\text{maximize}}\, V(D_1, \ldots, D_n) = \sum_{i=1}^{n} V_i\left(D_i\right). \tag{30}$$

This control objective has the same separable structure as (23); thus, we can tailor the algorithm in (28) and (29) to our setting. To this end, we discretize (28) and (29) using the forward rectangular rule, i.e.,

$$\dot{x} = \frac{x[k+1] - x[k]}{\delta}, \dot{z} = \frac{z[k+1] - z[k]}{\delta}. \tag{31}$$

Then, by defining $u = -\nabla \tilde{f}(x)$, and substituting it and the expression in (31) into (28) and (29), we obtain that

$$x[k+1] = (\mathbb{I}_{(n-1)^2} - \delta \tilde{L})x[k] - \delta \tilde{L} z[k] + \delta u[k]; \tag{32}$$

$$z[k+1] = z[k] + \delta \tilde{L} x[k]. \tag{33}$$

After introducing the tuning parameter γ to (32) and (33), we arrive at (13) and (14) as presented in Section II.

REFERENCES

[1] G. R. Walker and P. C. Sernia, "Cascaded DC-DC converter connection of photovoltaic modules," *IEEE Trans. Power Electron.*, vol. 19, no. 4, pp. 1130–1139, Jul. 2004.

[2] N. Femia, G. Lisi, G. Petrone, G. Spagnuolo, and M. Vitelli, "Distributed maximum power point tracking of photovoltaic arrays: Novel approach and system analysis," *IEEE Trans. Ind. Electron.*, vol. 55, no. 7, pp. 2610–2621, Jul. 2008.

[3] L. Linares, R. Erickson, S. MacAlpine, and M. Brandemuehl, "Improved energy capture in series string photovoltaics via smart distributed power electronics," in *Proc. Appl. Power Electron. Conf. Expo.*, 2009, pp. 904–910.

[4] R. Pilawa-Podgurski and D. Perreault, "Submodule integrated distributed maximum power point tracking for solar photovoltaic applications," *IEEE Trans. Power Electron.*, vol. 28, no. 6, pp. 2957–2967, Jun. 2013.

[5] Q. Li and P. Wolfs, "A review of the single phase photovoltaic module integrated converter topologies with three different dc link configurations," *IEEE Trans. Power Electron.*, vol. 23, no. 3, pp. 1320–1333, May 2008.

[6] R. Erickson and A. Rogers, "A microinverter for building-integrated photovoltaics," in *Proc. Appl. Power Electron. Conf. Expo.*, 2009, pp. 911–917.

[7] A. Trubitsyn, B. Pierquet, A. Hayman, G. Gamache, C. Sullivan, and D. Perreault, "High-efficiency inverter for photovoltaic applications," in *Proc. Energy Convers. Congr. Expo.*, 2010, pp. 2803–2810.

[8] P. Shenoy, K. Kim, B. Johnson, and P. Krein, "Differential power processing for increased energy production and reliability of photovoltaic systems," *IEEE Trans. Power Electron.*, vol. 28, no. 6, pp. 2968–2979, Jun. 2013.

[9] J. Stauth, M. Seeman, and K. Kesarwani, "Resonant switched-capacitor converters for sub-module distributed photovoltaic power management," *IEEE Trans. Power Electron.*, vol. 28, no. 3, pp. 1189–1198, Mar. 2013.

[10] C. Olalla, D. Clement, M. Rodriguez, and D. Maksimovic, "Architectures and control of submodule integrated dc-dc converters for photovoltaic applications," *IEEE Trans. Power Electron.*, vol. 28, no. 6, pp. 2980–2997, Jun. 2013.

[11] S. Qin and R. C. Pilawa-Podgurski, "Sub-module differential power processing for photovoltaic applications," in *Proc. Appl. Power Electron. Conf. Expo.*, 2013, pp. 101–108.

[12] S. Qin, A. J. Morrison, and R. C. Pilawa-Podgurski, "Enhancing microinverter energy capture with sub-module differential power processing," in *Proc. Appl. Power Electron. Conf. Expo*, 2014, pp. 621–628.

[13] C. Schaef and J. T. Stauth, "Multilevel power-point-tracking for variable-conversion-ratio photovoltaic ladder converters," in *Proc. Workshop Control Modeling Power Electron.*, 2013, pp. 1–7.

[14] S. Poshtkouhi, A. Biswas, and O. Trescases, "DC-DC converter for high granularity, sub-string mppt in photovoltaic applications using a virtual-parallel connection," in *Proc. Appl. Power Electron. Conf. Expo.*, 2012, pp. 86–92.

[15] T. Shimizu, M. Hirakata, T. Kamezawa, and H. Watanabe, "Generation control circuit for photovoltaic modules," *IEEE Trans. Power Electron.*, vol. 16, no. 3, pp. 293–300, May 2001.

[16] T. Shimizu, O. Hashimoto, and G. Kimura, "A novel high-performance utility-interactive photovoltaic inverter system," *IEEE Trans. Power Electron.*, vol. 18, no. 2, pp. 704–711, Mar. 2003.

[17] H. Bergveld, D. Buthker, C. Castello, T. Doorn, A. de Jong, R. van Otten, and K. de Waal, "Module-level dc/dc conversion for photovoltaic systems: The delta-conversion concept," *IEEE Trans. Power Electron.*, vol. 28, no. 4, pp. 2005–2013, Apr. 2013.

[18] H. Field and A. Gabor, "Cell binning method analysis to minimize mismatch losses and performance variation in SI-based modules," in *Proc. Photovoltaic Spec. Conf.*, 2002, pp. 418–421.

[19] C. E. Chamberlin, M. A. Rocheleau, M. W. Marshall, A. M. Reis, N. T. Coleman, and P. Lehman, "Comparison of PV module performance before and after 11 and 20 years of field exposure," in *Proc. Photovoltaic Spec. Conf.*, 2011, pp. 101–105.

[20] B. Gharesifard and J. Cortes, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Trans. Autom. Control*, vol. 59, no. 3, pp. 781–786, Mar. 2014.

[21] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[22] M. Villalva, J. Gazoli, and E. Filho, "Comprehensive approach to modeling and simulation of photovoltaic arrays," *IEEE Trans. Power Electron.*, vol. 24, no. 5, pp. 1198–1208, May 2009.

[23] R. Erickson and D. Maksimovic, *Fundamentals of Power Electronics*. Norwell, MA, USA: Kluwer, 2000.

[24] R. J. Serna, B. J. Pierquet, J. Santiago, and R. Pilawa-Podgurski, "Field measurements of transient effects in photovoltaic panels and its importance in the design of maximum power point trackers," in *Proc. Appl. Power Electron. Conf. Expo.*, 2013, pp. 3005–3010.

[25] C. Deline, B. Marion, J. Granata, and S. G. Sigifredo, "A performance and economic analysis of distributed power electronics in photovoltaic systems," Nat. Renewable Energy Lab., Golden, CO, USA, Tech. Rep. NREL/TP-5200-50003, Jan. 2011.

[26] A. Peterchev and S. Sanders, "Quantization resolution and limit cycling in digitally controlled PWM converters," *IEEE Trans. Power Electron.*, vol. 18, no. 1, pp. 301–308, Jan. 2003.

[27] C. Barth and R. Pilawa-Podgurski, "Implementation of dithering digital ripple correlation control," in *Proc. Workshop Control, Modeling Simul. Power Electron.*, 2013, pp. 1–7.

[28] C. Barth and R. Pilawa-Podgurski, "Dithering digital ripple correlation control with digitally-assisted windowed sensing for solar photovoltaic mppt," in *Proc. Appl. Power Electron. Conf. Expo.*, 2014, pp. 1738–1746.

[29] S. Qin, K. Kim and R. Pilawa-Podgurski, "Laboratory emulation of a photovoltaic module for controllable insolation and realistic dynamic performance," in *Proc. Power Energy Conf. Illinois*, 2013, pp. 23–29.

Shibin Qin (S'12) received the B.E. degree in electrical engineering from Huazhong University of Science and Technology, Wuhan, China, in 2012 and the M.S. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2014, where he is currently working toward the Ph.D. degree.

His research is in power electronics in photovoltaic applications.

Stanton T. Cady (S'10) received the B.S. degree in electrical engineering in 2009 and the M.S. degree in electrical engineering in 2011, both from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, where he is currently working toward the Ph.D. degree in electrical engineering.

His research interests include distributed control and its application to electric power systems and power electronics.

Alejandro D. Domínguez-García (S'02—M'07) received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2007 and the degree of electrical engineer from the University of Oviedo, Oviedo, Spain, in 2001.

After finishing the Ph.D. degree, he spent some time as a Postdoctoral Research Associate at the Laboratory for Electromagnetic and Electronic Systems, Massachusetts Institute of Technology. He is currently an Assistant Professor at the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Champaign, IL, USA, where he is affiliated with the power and energy systems area. His research interests include the areas of system reliability theory and control, and their application to electric power systems, power electronics, and embedded electronic systems for safety-critical/fault-tolerant aircraft, aerospace, and automotive applications.

Dr. Domínguez-García received the National Science Foundation CAREER Award in 2010, and the Young Engineer Award from the IEEE Power and Energy Society in 2012. He has also been a Grainger Associate since 2011. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON POWER SYSTEMS and the IEEE POWER ENGINEERING LETTERS.

Robert Carl Nikolai Pilawa-Podgurski (S'06—M'11) was born in Hedemora, Sweden. He received dual B.S. degrees in physics, electrical engineering, and computer science in 2005, the M.Eng. degree in electrical engineering and computer science in 2007, and the Ph.D. degree in electrical engineering in 2012, all from the Massachusetts Institute of Technology, Cambridge, MA, USA.

He is currently an Assistant Professor at the Electrical and Computer Engineering Department, University of Illinois, Urbana-Champaign, Champaign, IL, USA, and is affiliated with the Power and Energy Systems Group. He performs research in the area of power electronics. His research interests include renewable energy applications, energy harvesting, CMOS power management, and advanced control of power converters.

Dr. Pilawa-Podgurski received the Chorafas Award for outstanding MIT EECS Master's thesis, the Google Faculty Research Award, and the Richard M. Bass Outstanding Young Power Electronics Engineer Award of the IEEE Power Electronics Society. He is an coauthor of two IEEE prize papers.