Deep Learning-enhanced Block-Diagram Modeling of Solar Power Systems

Thomas E. McDermott
Meltran, Inc.
Charlottesville, VA, USA
0000-0001-5496-0419

Qiuhua Huang
CO School of Mines
Golden, CO, USA
0000-0001-8457-7429

Yuan Liu
Pacific NW Nat. Lab.
Richland, WA, USA
0000-0001-5137-3040

Daniel Glover
WA State Univ.
Pullman, WA, USA
0000-0003-3896-7631

Meghana Ramesh
Pacific NW Nat. Lab.
Richland, WA, USA
0000-0002-2113-4038

Kenneth McDonald
Univ. Central FL
Orlando, FL, USA
0009-0008-1128-640X

Ganesh Marasini
Univ. Central FL
Orlando, FL, USA
0009-0009-2149-9537

Zhihua Qu
Univ. Central FL
Orlando, FL, USA
0000-0001-6710-7134

***Abstract*—Data-driven models of power system inverter-based resources are desired to run simulations faster than with detailed electromagnetic transient models, to hide proprietary design details, to support control system design applications, and to aggregate the effects of distributed energy resources. This paper applies a customized Hammerstein Wiener framework to train block diagram models from thousands of electromagnetic transient simulations or experimental test records. The block diagram models integrate with larger grid simulations as voltage-controlled current sources or current-controlled voltage sources for several simulators. Guidelines for block architecture and training are presented. Three-phase balanced, three-phase unbalanced, and single-phase examples all achieve an acceptable root mean square error of no more than 0.05 per-unit.**

***Index Terms*—Deep learning, inverters, photovoltaic systems, power system transients, system identification**

I. Introduction

Renewable energy sources for the electric power system include primarily solar photovoltaic and wind generation, supplemented with battery energy storage systems to serve load through solar and wind power variations. Solar, storage, and many wind generators connect to the grid through power electronic converters, so they are referred to as inverter-based resources (IBR). Some IBR can operate in grid-forming mode (GFM), establishing grid voltage and frequency and providing grid restoration service. This paper focuses on solar IBR operating as GFM, but the approach also applies to grid-following (GFL) mode, battery energy storage, and wind generation with full-converter interface.

Conventional rotating machine generators (fossil, nuclear, hydro) are represented with well-understood equivalent circuit models in a *dq0* reference frame [1]. They are parameterized from standard vendor test reports and implemented in all practical dynamic or electromagnetic transient (EMT) simulators used for grid studies. Data-driven models are possible but not required for these generators; the *dq0* equivalent circuits have been adequate for both dynamic and EMT studies.

For IBRs, some generic models exist for dynamic studies, but their limitations in accuracy have resulted in a strong preference by the North American Electric Reliability Corporation for its regulated electric utilities to adopt EMT studies for IBR [2]. EMT simulators enable the most detailed IBR studies, but they take a long time to simulate, especially the ”switching models” that represent power electronic switching operations at high frequency. The ”average models” run faster but may compromise accuracy. Another drawback of EMT is that IBR models have not been portable among different simulators, creating a burden of adopting EMT methods due to software licensing costs, training, and model translation labor. Some IBR vendors have been reluctant to provide EMT models because of the potential of exposing proprietary design details, but this is changing due to the development of a dynamic link library (DLL) interface, under Cigre/IEEE auspices, for ”real-code” IBR modeling. Many EMT vendors are supporting this DLL interface, improving model portability.

Data-driven models offer some advantages over detailed EMT models:

- They can hide proprietary design details of the inverter from the model user.
- Simulation times are much faster, especially compared to switching EMT models.
- The data-driven model behavior is nonlinear, but continuous and differentiable. These properties are helpful to control system design and simulation.
- Data-driven models can be implemented in library functions or “blocks” in the end-user’s EMT simulator.

An EMT simulator produces transient output data, sometimes called “waveforms” or “point-on-wave” data. Laboratory

This material is based upon work supported by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number DE-EE0009028. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

tests of IBR also produce transient output data [3]. A data-driven model could be trained from any sources of transient output data, e.g., from combinations of EMT outputs, laboratory test data, and/or field event records. A data-driven model can be trained even if no EMT model is available. This paper includes examples of training from EMT and lab data.

The general options for data-driven modeling are summarized in [4]. The Hammerstein Wiener (HW) modeling framework has been applied to active distribution systems [5], inverter short-circuit contributions [3], and photovoltaic panels [6]. Another black-box modeling approach was applied to inverter dynamics in [7]. The MATLAB System Identification (SI) Toolbox provides HW and non-linear autoregressive (NARX) model fitting tools that are widely available to researchers. In [8], a customized artificial neural network (ANN) was found to outperform the SI Toolbox HW and NARX options. In the literature so far, good model fitting accuracy has been reported, but with small datasets of two [8] to forty-four [9] events. Some difficulties have arisen with integration of the data-driven models in system simulations [9], where the model attempts to fit transient waveforms including harmonics.

We started with SI Toolbox functions to choose the HW framework over others, but then needed to scale up the problem size, customize the loss functions, and train the models faster. We settled on *dynoNet* [10] because:

- It supports non-linear, causal dynamic relations, such as series, parallel and feedback blocks with multiple and output (MIMO) structures.
- It models each block as a neural network layer, and trains them end-to-end by plain back-propagation, using the deep learning framework *PyTorch* [11].
- Training was several times faster than with SI Toolbox.
- *dynoNet* can guarantee open-loop model stability in the discrete time domain.
- We could customize the HW framework for this project, as [8] did with ANN.

This paper presents generalized block diagram models in the HW framework for Photovoltaics (HWPV). Its main contributions are:

- Examples include single-phase and three-phase inverters, trained from EMT and lab data, covering GFM and GFL applications, with average and switching models.
- Fitting in the *dq0* frame mitigated some issues from [9].
- Guidelines are presented for model architecture, hyperparameters, input feature selection, and data preparation.
- *dynoNet* has been improved with a *PyPi* installer, and code fixes for the stable second-order dynamic block option. Distinct real poles for this option have been implemented with equation (32) of [10].
- HWPV simulation (forward evaluation) has been implemented in five platforms, including MATLAB/Simulink.
- Source code, examples, documentation [12], and datasets [13] have been published open-source.

II. Model Architecture

Fig. 1 shows the functional characteristics of solar IBR to be modeled, including nonlinear photovoltaic (PV) array, the DC/AC inverter, and the inductor-capacitor-inductor (LCL) filter. The DC/AC inverter may comprise a switching or average model of the power electronic devices, it may include a separate DC/DC conversion stage, and it typically includes some internal passive filter components. It connects to the grid at a point of common coupling (PCC). The important external control parameters for GFM include the *dq* voltage control indices, U_d and U_q, and the control frequency, F_c. Other external control signals include *Ctl* to indicate startup, GFM, or GFL mode, and *Unb* to indicate balanced or unbalanced operation. *Ctl* and *Unb* are training inputs to improve the model accuracy. In application, these signals could come from separate device functions, e.g., a negative sequence relay, ground fault relay, or a mode transition command. Other external control and protection functions, e.g., under/overvoltage trip or under/overfrequency trip, can be added outside of the HWPV model. Internal protection functions of the DC/AC converter, i.e., between the DC and O nodes, would have to be included in the HWPV model.

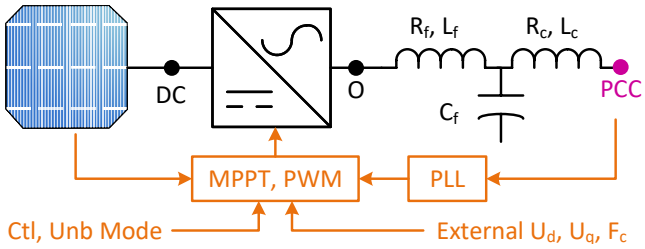

Fig. 1. Functional diagram of solar IBR characteristics in the HWPV model.

Reference [14] shows the importance of representing phase locked loop (PLL), pulse width modulation (PWM) and maximum power point tracking (MPPT) in a detailed EMT model of a single-phase inverter. In some public examples, the PV array model does not exhibit a variable peak power point for the MPPT to adjust, so we adopted the model in [15] for EMT simulation, with an incremental conductance MPPT implementation from [16]. In GFL mode, the PLL is necessary to implement a *dq0* transformation of voltages and currents. In GFM mode, the control frequency, F_c, defines the *dq0* transformation via (1) and (2). A controlled current source (Norton equivalent) will inject phase currents at the PCC according to (3). The controlling phase voltages transform to *dq0* components according to (4). The root mean square (RMS) voltage, used to create a polynomial input feature to the HWPV model, is defined in (5) for three-phase inverters. For single-phase inverters, the factor 1.5 becomes 1.0 in (5), and the second order generalized integrator (SOGI) and frequency locked loop (FLL) [17] produce the *dq* signals.

$$\theta = \theta_o + \int_{t_0}^{t} 2\pi F_c dt \tag{1}$$

$$\mathbf{T_{dq0}} = \frac{2}{3}\begin{bmatrix} \cos(\theta) & \cos(\theta - \frac{2\pi}{3}) & \cos(\theta + \frac{2\pi}{3}) \\ -\sin(\theta) & -\sin(\theta - \frac{2\pi}{3}) & -\sin(\theta + \frac{2\pi}{3}) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2)$$

$$\mathbf{I}_\phi = \mathbf{T_{dq0}}^{-1} \begin{bmatrix} I_d & I_q & 0 \end{bmatrix}^T \quad (3)$$

$$\mathbf{V_{dq0}} = \mathbf{T_{dq0}} \begin{bmatrix} V_a & V_b & V_c \end{bmatrix}^T \quad (4)$$

$$V_{rms} = \sqrt{1.5({V_d}^2 + {V_q}^2)} \quad (5)$$

Fig. 2 shows the trained HWPV blocks and a controlled Norton current source connected to the grid at the PCC. The **F1** and **F2** blocks are static nonlinear functions ($tanh$ activation) implemented in a neural network. The $\mathbf{H1}(\mathbf{z})$ block is a linear dynamic operator. The input vector, $\mathbf{u_b}$, includes only information available at the IBR:

- Weather inputs solar irradiance, G, and temperature, T.
- External control inputs U_d, U_q, Ctl, Unb, and F_c.
- Transformed voltages, V_d and V_q, from $\mathbf{V}_\phi$ at the PCC. Use of V_{rms} produced less accurate HWPV models.
- A polynomial input feature, GV_{rms}, which is available from other inputs and provides information about the expected IBR power output. This polynomial feature improves fitting accuracy and model stability, especially during startup ($Ctl = 0$).

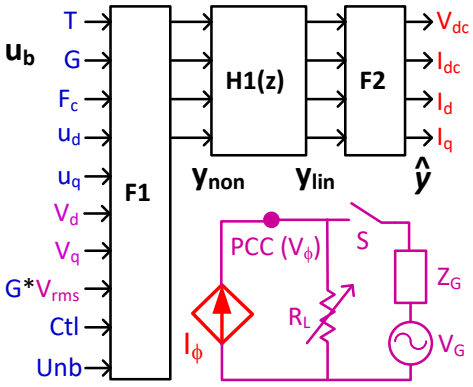

Fig. 2. Block diagram model and controlled current source with grid interface.

The output vector, $\hat{\mathbf{y}}$, includes:

- DC voltage, V_{dc}, and current, I_{dc}. These are control system inputs in [18], but not connected to a DC circuit.
- AC currents, I_d and I_q, for injecting $\mathbf{I}_\phi$ (3) at the PCC.

The switch, S, is open for GFM and closed for GFL. The grid source equivalent (Z_G, V_G) and load (R_L) affect $\mathbf{u_b}$ indirectly by relating $\mathbf{V}_\phi$ and $\mathbf{I}_\phi$ at PCC. If the GFM load is purely resistive, as shown in Fig. 2, the *dq* axes are decoupled, with voltages and currents related as in (6). Otherwise, the *dq* axes are coupled through the LCL filter and any reactive components of the load and grid impedances.

$$V_d = I_d R_L; V_q = I_q R_L \quad (6)$$

The HWPV model fitting did not improve with the addition of more blocks, e.g., a six-block cascade to separate the AC and DC stages, or a four-block structure with feedback. The basic three-block structure in Fig. 2 produced the best results, allowing *dynoNet* and *PyTorch* to optimize the parameters.

III. Training and Validation

HWPV training requires hundreds or thousands of transient event records from simulation, lab and/or field testing. A portion of these is held back for validation, and the others are used for training. The basic loss function is normalized root mean square error (RMSE) over $\hat{\mathbf{y}}$, to be < 0.05.

Table I summarizes the EMT models and hardware specimens used to fit HWPV models. Consortium for Electric Reliability Technology Solutions (CERTS) MicroGrid inverter is three-phase, 480 V, 100 kW, and GFM [19]. The *GridLink* is a three-phase, 400 V, 5 kW, GFM, software defined inverter (SDI) from Siemens. The *OSG4* is a single-phase, 240 V, 13.5 kW inverter derived from an Alternative Transients Program (ATP) [20] example, enhanced for GFM operation. The $Lab2$ specimen is a vendor-supplied single-phase, 240 V, 8 kW GFM inverter; it has no access to internal measurement points. Δt is the EMT simulation time step or instrumentation sample interval. N_c is the number of transient records collected.

Fig. 3 shows the *GridLink* SDI connected to load, grid, and instrumentation for data collection. The *GridLink* is controllable and provides internal measurements, but the instrumentation in Fig. 3 supports higher resolution and lower Δt.

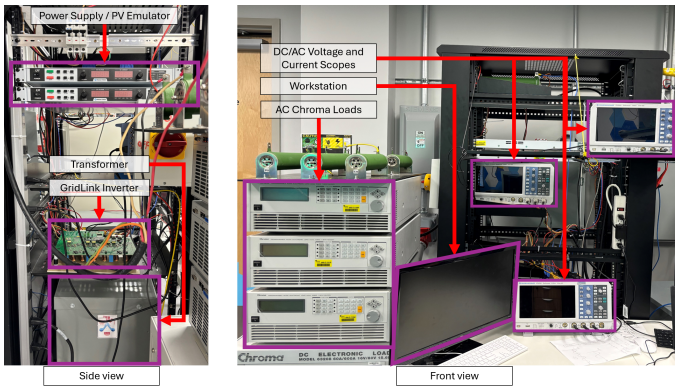

Fig. 3. Testing and data acquisition systems for the GridLink SDI.

Table II summarizes the input and output signals used for HWPV model fitting. Some of the signals were decimated with low-pass, second-order, zero-shift Butterworth filters before assembling transient data sets. These include V_d, V_q, I_d, I_q, and I_{dc}, except in the cases of *UCF2*, *UCF3*, and *UCF4**, which already had comparable filtering in the MATLAB simulation. Any EMT effects on these channels would appear in the HWPV model as filtered components.

The **H1** block maintains internal state, which should be initialized for the HWPV model to start up correctly. Bias coefficients in **F1** produce non-zero initial conditions on **H1** even when $\mathbf{u_b}$ is zero. In training, this was accomplished by pre-padding $\mathbf{u_b}$ with about 500 points of initial values, then ignoring half the pre-padded region in loss evaluation. In forward evaluation, initialization of $\mathbf{H1}(\mathbf{z})$ is determined from

TABLE I
INVERTER TEST SYSTEM CHARACTERISTICS, HYPERPARAMETERS AND RESULTING OUTPUT CHANNEL ACCURACY

Model	**Description**	$\mathbf{\Delta t}$	$\mathbf{N_c}$	$\mathbf{z\Delta t}$	$\mathbf{N_1}$	$\mathbf{N_2}$	$\mathbf{N_A}$	$\mathbf{N_B}$	$\mathbf{N_b}$	$\mathbf{P_v}$	**lr**	ϵ	$\mathbf{N_i}$	**Hrs**[a]	**Normalized RMSE**				$\mathbf{\% > 0.05}$
															V_{dc}	I_{dc}	I_d	I_q	
Bal3	CERTS Avg. (ATP)	10e-6	23,400	0.002	80	60	2	2	20	10	1e-3	1e-6	1000	37.14	0.0086	0.0074	0.0053	0.0012	0.026
OSG4	$1-\phi$ Swt. (ATP)	0.5e-6	380	0.001	80	60	4	4	19	15	5e-4	1e-8	2000	2.87	0.0172	0.0082	0.0056	0.0027	0.0
UCF2	SDI (MATLAB)	1e-6	1,500	0.002	80	60	2	2	50	15	1e-3	1e-8	2000	3.00	0.0018	0.0026	0.0013	0.0017	0.0
UCF3	SDI (MATLAB)	1e-6	2,100	0.010	80	60	2	2	105	15	1e-3	1e-8	1000	0.43	0.0030	0.0116	0.0111	0.0047	0.0
UCF4n	SDI (MATLAB)	1e-6	8,100	0.010	80	60	2	3	405	15	1e-3	1e-8	1000	1.78	0.0084	0.0321	0.0356	0.0161	12.42
UCF4nz	SDI (MATLAB)	1e-6	8,100	〃	〃	〃	〃	〃	〃	〃	〃	〃	〃	1.47	0.0070	0.0307	0.0329	0.0161	7.90
															V_{dc}	I_{dc}	V_d	V_q	
UCF4t	SDI (MATLAB)	1e-6	8,100	0.010	80	60	2	3	405	15	1e-3	1e-8	1000	1.97	0.0161	0.0169	0.0184	0.0092	0.0
UCF4tz	SDI (MATLAB)	1e-6	8,100	〃	〃	〃	〃	〃	〃	〃	〃	〃	〃	1.52	0.0090	0.0028	0.0072	0.0031	0.0
															V_{dc}	I_{dc}	I_{dlo}	I_{qlo}	
															0.0037	0.0015	0.0023	0.0016	
															I_{0lo}	I_{dhi}	I_{qhi}	I_{0hi}	
Unb3	CERTS Avg. (ATP)	10e-6	2,430	0.002	120	60	2	2	27	10	1e-3	1e-8	1000	12.30	0.0022	0.0038	0.0040	0.0018	0.0
															I_{dc}	I_d	I_q		
SDI5	SDI (Test)	5e-5	288	0.001	60	40	2	2	18	19	1e-3	1e-6	5000	0.77	0.0278	0.0278	0.0406		8.33
															V_{rms}	I_{rms}	I_{dc}		
Lab2	$1-\phi$ GFM (Test)	1.2e-4	69	1.2e-4	30	30	2	2	10	10	1e-3	1e-8	1000	0.06	0.0068	0.0252	0.0276		0.0

[a]With Intel i7-13700K, 64 GB RAM, and Nvidia RTX 4070 Ti card.

TABLE II
INVERTER MODEL TRANSIENT INPUT CHANNELS

Model	**Inputs**
Bal3	$G, T, F_c, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$
OSG4	$G, T, F_c, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$
UCF2	$G, T, F_c, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$
UCF3	$G, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$
UCF4n	$G, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$
UCF4t	$G, U_d, U_q, Ctl, I_d, I_q, GI_{rms}$
Unb3	$G, T, F_c, U_d, U_q, Ctl, Unb, V_{dlo,hi}, V_{qlo,hi}, V_{0lo,hi}, GV_{rmslo}$
SDI5	$F_c, U_d, U_q, V_d, V_q, V_{dc}$
Lab2	F_c, V_c, R_c, V_{dc}

the known initial output of $\mathbf{F1}$ ($\mathbf{y_{non}}$ in Fig. 2), using (7) on each MIMO pairing of output, m, to input, p [10].

$$H_{1(m,p)}(z=1) = \frac{b_0 + b_1 + b_2 + b_3 ...}{1 + a_1 + a_2 + a_3 ...} \tag{7}$$

Table I includes the hyperparameters used for each HWPV model, as tuned by cut-and-try:

- $z\Delta t$ is the decimated time step for $\mathbf{H1(z)}$.
- N_1 is the number of $tanh$ hidden cells in $\mathbf{F1}$. It should be proportional to the number of inputs squared.
- N_2 is the number of $tanh$ hidden cells in $\mathbf{F2}$. It should be proportional to the number of outputs squared.
- N_A is the number of learnable denominator coefficients for each MIMO pair in $\mathbf{H1(z)}$, plus $a_0 = 1$.
- N_B is the number of learnable numerator coefficients in $\mathbf{H1(z)}$, including b_0. N_A and N_B range from 2 to 8, using the higher orders with EMT switching models.
- N_b is the batch size, typically no more than 10% of N_c.
- P_v is the percentage of records reserved for validation, typically 10 to 20.
- lr is the learning rate, usually 1e-3, but smaller for EMT switching models.
- ϵ is a numerical stability constant for the optimizer [11].
- N_i is the number of training epochs for RMSE < 0.05.

The transient datasets should cover the expected range of HWPV model operation, including startup, GFM, and GFL modes. For example, to collect 23,400 transient records for the $Bal3$ model, the EMT simulation starts to one of 360 steady-state conditions, as combinations of:

- T set at one of 2 values in [15.0, 35.0] C.
- G ramps from 0 to one of 10 values in the set [100 to 1000, step 100] W/m^2.
- F_c set at 1 value in [60.0] Hz.
- U_d set at one of 2 values in [0.95, 1.0] pu.
- U_q set at one of 3 values in [-0.05, 0.001, 0.05] pu.
- R_L set for one of 3 power P_{pu} values in [0.95, 1.00, 1.05] per (8); $P_{nom} = 100e3$ W and $V_{nom} = 480$ V

$$R_L = \frac{V_{nom}^2}{P_{nom}} \frac{1000}{G} \frac{1}{P_{pu}} \tag{8}$$

The startup ramp in G occurs from 1.0 to 2.0 s. Ctl changes from 0.0 to 1.0 at 2.0 s. At 3.0 s, one of the following 65 input changes occurs, and the simulation continues to 5.0 s.

- T steps to 1 other value in [15.0, 35.0]
- G ramps over 1 s to one of 4 other values defined in a lookup table, approximately ± 100 and ± 200, but shifted to maintain $0 \leq G \leq 1000$ W/m^2
- F_c steps by 1 of 10 values in [-5 to 5, step 1, but not 0]
- U_d steps by 1 of 10 values in [-0.20 to -0.02, step 0.02]
- U_q steps by 1 of 20 values [-0.5 to 0.5, step 0.05, not 0]
- R_L divisor 1 of 20 values [0.8 to 1.2, step 0.02, not 1]

Training data sets for all models in Table I have been plotted and archived [13] for others to use.

In (4), the transformed *0* components are zero except under unbalanced conditions, when $Unb = 1$. The transformation of unbalanced voltages in (4) will produce oscillating *dq* components rather than constant values. These voltages are pre-processed through high-pass and low-pass 6th-order Butterworth filters, 0.05 critical frequency, to resolve them into their predominant DC and 120 Hz *dq* or 60 Hz *0* components.

The low-pass components are approximately constant, and the moving averages of the high-pass components are also approximately constant. The HWPV model then fits output *dq0* current separately as I_{dlo}, I_{qlo}, I_{0lo}, I_{dhi}, I_{qhi}, and I_{0hi}. The total *dqo* currents are constructed from (9), (10), and (11). These *dq0* currents oscillate, but (3) still applies to inject the phase currents at PCC. Fig. 4 shows a sample set of these filtered components used to train the *Unb3* model.

$$I_d = I_{dlo} + I_{dhi}\sin(4\pi F_c t) \tag{9}$$

$$I_q = I_{qlo} + I_{qhi}\sin(4\pi F_c t) \tag{10}$$

$$I_0 = I_{0lo} + I_{0hi}\sin(2\pi F_c t) \tag{11}$$

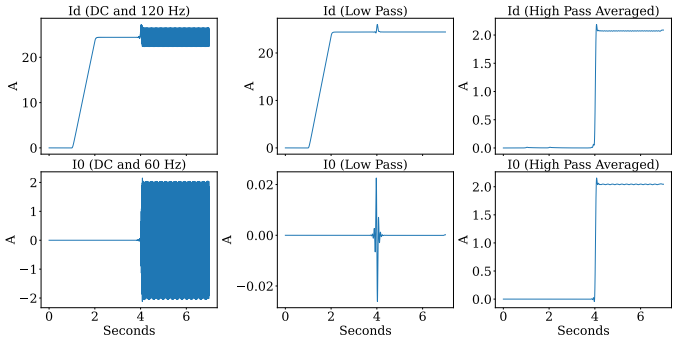

Fig. 4. Filtered unbalanced *dq0* voltage and current components.

IV. Model Accuracy and Application

Table I includes model training times in hours, Hrs, and the RMSE metrics. RMSE captures in a single metric any errors in transient peaks, steady-state values, event start times, settling times, damping rates, etc. The right-most column shows percentage of records with RMSE > 0.05. Fig. 5 shows a typical accuracy comparison for I_d and I_q, which are the most important outputs in a Norton interface to the grid. Individual records may have higher RMSE values, as in Fig. 6, which is the worst result in I_d for the *Bal3* model. Even though the overall RMSE for I_d is 0.0053, it's 0.0559 for this case, and 6 of the 23,400 records have RMSE > 0.05 in I_d. The error looks higher in Fig. 6 because RMSE is normalized to the whole range of I_d, which is 0 to 202 in the *Bal3* model. The maximum true value (y) of I_d in Fig. 6 is about 55. Furthermore, the true and estimated values only deviate from each other over the time span from 4 to 7 s. An external controller should act to reduce any steady-state error in the expected IBR output.

The overall normalized RMSE values in Table I are all ≤ 0.05. However, models *UCF4n*, *UCF4nz*, and *SDI5* have too many outliers > 0.05 in the last column. More training data should help *SDI5*'s accuracy. The Thevenin model *UCF4t*, discussed later, provides better accuracy than *UCF4n*.

Some control applications may require the voltage at node O instead of the PCC in Fig. 1. Non-causal phasor arithmetic defines the voltages and currents at O from conditions at PCC

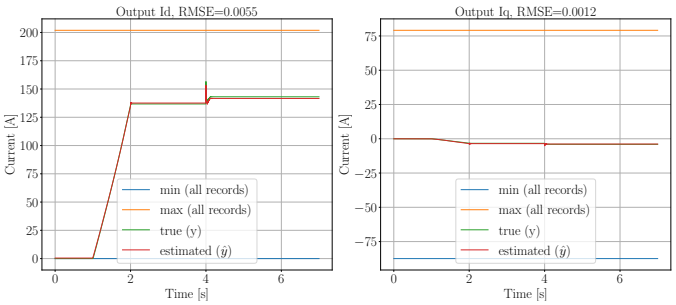

Fig. 5. Typical RMSE for I_d and I_q in *Bal3* model.

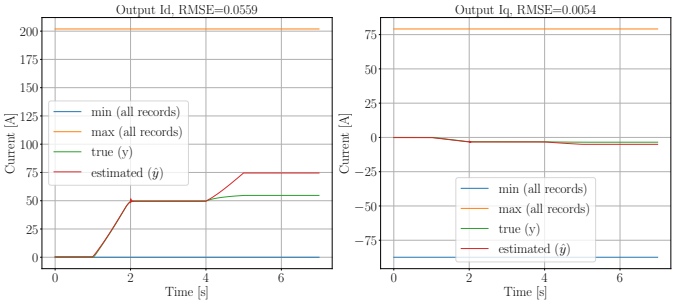

Fig. 6. Worst RMSE for I_d in *Bal3* model; I_d reaches 27.1% of its maximum.

and LCL parameters, but a causal approach is provided in [18]. Transient data from O may not be available or suitable for model training. Fig. 7 (left) shows a noisy voltage source converter (VSC) voltage at point O in a single-phase switching EMT model. The voltage at the LCL Terminals (PCC) is passed through a low-pass filter ($\tau = 0.0002$) for processing through the SOGI and FLL. These processed voltages, used for HWPV training, are shown in Fig. 7 (right).

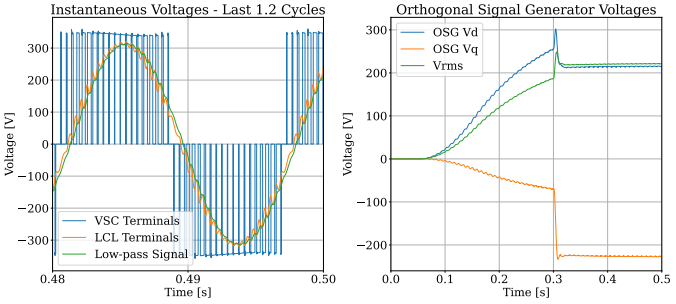

Fig. 7. Voltage outputs from 1-phase PWM inverter model *OSG4* in ATP.

The HWPV model coefficients are exported into a human-readable Javascript Object Notation (JSON) format, with forward evaluation implemented so far in:

- Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS) [21], supporting some EMT tools.
- A Python test harness emulating variable grid impedance.
- Lightweight Raspberry Pi implementation with *numpy*.
- ATP's TACS feature, which does not support vector arithmetic and allows only 1000 local variables. However, ATP's MODELS feature was not robust.

- A native MATLAB/Simulink implementation.

A DLL version is under development for more EMT tools.

V. Stability Analysis

The open-loop poles of $\mathbf{H1}(\mathbf{z})$ are stable in the z domain, guaranteed when *dynoNet* trains a cascade of stable 2nd-order blocks and 1st-order blocks. However, it's insufficient to meet RMSE acceptance criteria with stable $\mathbf{H1}(\mathbf{z})$. The trained model must also be stable in forward evaluation when connected to a grid, in a closed loop, which doesn't always occur in weak interconnections.

The Norton controlled current source in Fig. 2 is compatible with many grid system dynamic and EMT solvers that use a nodal admittance formulation. However, disturbances in I_ϕ, injected into R_L, create disturbances in V_d and V_q at the inputs to $\mathbf{F1}$. This closes a feedback loop, and the constraint (12) must hold for stable operation of the Norton model. For Thevenin models, invert each fraction in (12).

$$\sigma = max(\frac{\partial I_d}{\partial V_d}, \frac{\partial I_d}{\partial V_q}, \frac{\partial I_q}{\partial V_d}, \frac{\partial I_q}{\partial V_q}) < \frac{1}{R_L} \tag{12}$$

Table III shows σ obtained with a weighted addition of (12) to the training loss function for *UCF4n* and *UCF4t*. With no σ optimization in *UCF4nz* and *UCF4tz*, the RMSE is lower in Table I, but σ exceeds the target values of 0.01 and 70.0, respectively. With σ optimization, only the Thevenin model *UCF4t* satisfies the constraints on both σ and RMSE, because both Norton models have too many cases with RMSE > 0.05. It takes about 30% more time to train with σ optimization.

TABLE III
Thevenin and Norton Model Sensitivities

Source	Model	Weight	$\sigma_{\mathbf{targ}}$	$\sigma_{\mathbf{act}}$
Norton	UCF4n	0.0010	0.01	0.1621
Norton	UCF4nz	None	None	0.3830
Thevenin	UCF4t	0.0035	70.0	13.92
Thevenin	UCF4tz	None	None	163.57

The trained $\mathbf{H1}(\mathbf{z})$ implements a fixed $z\Delta t$ as listed in Table I. For efficiency, usability, and compatibility with other models in a grid system simulation, variations in Δt must be allowed in forward evaluation of the HWPV model. We used the Python packages *control* [22] and *harold* [23] to convert $\mathbf{H1}(\mathbf{z})$ to $\mathbf{H1}(\mathbf{s})$. The Forward Euler method in *harold.undiscretize* produced stable poles in $\mathbf{H1}(\mathbf{s})$, while Backward Euler and Bilinear sometimes did not. The *dynoNet* stable second order function option for $\mathbf{H1}(\mathbf{z})$, where $N_A = 2$ and $N_B = 3$, has not produced unstable poles in $\mathbf{H1}(\mathbf{s})$.

VI. Conclusion

It has been shown that HWPV models of IBR can meet RMSE acceptance criteria for a variety of applications, given sufficient attention to data collection, model architecture, and input data preparation. The HWPV models integrate with grid system simulations, provided that constraints on output current (or voltage) sensitivities to input voltage (or current) disturbances are met. Future work includes distributed energy resource aggregation and variable continuous time stepping.

References

[1] J. Martinez, B. Johnson, and C. Grande-Moran, "Parameter determination for modeling system transients - part iv: rotating machines," *IEEE Transactions on Power Delivery*, vol. 20, no. 3, pp. 2063–2072, 2005.

[2] NERC, "Electromagnetic Transient Modeling Task Force," https://www.nerc.com/comm/RSTC/Pages/EMTTF.aspx, 2023.

[3] L. M. Wieserman, S. F. Graziani, T. E. McDermott, R. C. Dugan, and Z. Mao, "Test-based modeling of photovoltaic inverter impact on distribution systems," *IEEE 46th Photovoltaic Specialists Conference (PVSC)*, 2019.

[4] J. Schoukens and L. Ljung, "Nonlinear system identification: A user-oriented roadmap," https://arxiv.org/abs/1902.00683, 2019.

[5] L. D. P. Ospina, V. U. Salazar, and D. P. Ospina, "Dynamic equivalents of nonlinear active distribution networks based on Hammerstein-Wiener models: An application for long-term power system phenomena," *IEEE Transactions on Power Systems*, vol. 37, no. 6, pp. 4286–4296, 2022.

[6] A. Alqahtani, S. Marafi, B. Musallam, and N. El Din Abd El Khalek, "Photovoltaic power forecasting model based on nonlinear system identification," *Canadian Journal of Electrical and Computer Engineering*, vol. 39, no. 3, pp. 243–250, 2016.

[7] V. Valdivia, A. Lazaro, A. Barrado, P. Zumel, C. Fernandez, and M. Sanz, "Black-box modeling of three-phase voltage source inverters for system-level analysis," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 9, pp. 3648–3662, 2012.

[8] L. Qiao, Y. Xue, Y. Gui, W. Du, and F. F. Wang, "Comparative study of nonlinear black-box modeling for power electronics converters," in *2023 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2023, pp. 1446–1452.

[9] E. Kaufhold, S. Grandl, J. Meyer, and P. Schegner, "Feasibility of black-box time domain modeling of single-phase photovoltaic inverters using artificial neural networks," *Energies*, vol. 14, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/8/2118

[10] M. Forgione and D. Piga, "*dynoNet*: A neural network architecture for learning dynamical systems," *International Journal of Adaptive Control and Signal Processing*, vol. 35, no. 4, pp. 612–626, 2021.

[11] "Pytorch," https://pytorch.org/docs/stable/index.html/, 2024.

[12] T. E. McDermott, "pecblocks documentation," https://pecblocks.readthedocs.io/en/latest/, 2023.

[13] T. McDermott, "Replication data for: Deep learning-enhanced block-diagram modeling of solar power systems," 2024. [Online]. Available: https://doi.org/10.7910/DVN/PWW981

[14] M. E. Ropp and S. Gonzalez, "Development of a MATLAB/Simulink model of a single-phase grid-connected photovoltaic system," *IEEE Transactions on Energy Conversion*, 2009.

[15] M. G. Villalva, J. R. Gazoli, and E. R. Filho, "Comprehensive approach to modeling and simulation of photovoltaic arrays," *IEEE Transactions on Power Electronics*, vol. 24, no. 5, pp. 1198–1208, 2009.

[16] B. Subudhi and R. Pradhan, "A comparative study on maximum power point tracking techniques for photovoltaic power systems," *IEEE Transactions on Sustainable Energy*, vol. 4, no. 1, pp. 89–98, 2013.

[17] C. Zhang, S. Føyen, J. A. Suul, and M. Molinas, "Modeling and analysis of sogi-pll/fll-based synchronization units: Stability impacts of different frequency-feedback paths," *IEEE Transactions on Energy Conversion*, vol. 36, no. 3, pp. 2047–2058, 2021.

[18] G. Marasini, K. McDonald, Z. Qu, and T. E. McDermott, "Data-driven dynamic model and model reference control of inverter based resources," symposium on Systems Theory in Data and Optimization (SysDO), October 2024.

[19] W. Du, R. H. Lasseter, and A. S. Khalsa, "Survivability of autonomous microgrid during overload events," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3515–3524, 2019.

[20] European EMTP Users Group, "Alternative Transients Program (ATP-EMTP)," http://www.atp-emtp.org/, 2020.

[21] T. D. Hardy, B. Palmintier, P. L. Top, D. Krishnamurthy, and J. C. Fuller, "Helics: A co-simulation framework for scalable multi-domain modeling and analysis," *IEEE Access*, vol. 12, pp. 24 325–24 347, 2024.

[22] S. Fuller, B. Greiner, J. Moore, R. Murray, R. van Paassen, and R. Yorke, "The python control systems library (python-control)," in *60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 4875–4881.

[23] I. Polat, "An open-source systems and controls toolbox for python3," https://github.com/ilayn/harold, 2018.