

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

Thomas E. McDermott¹*, QiuHua Huang², Yuan Liu³, Daniel Glover⁴, Meghana Ramesh³, Kenneth McDonald⁵, Ganesh Marasini⁵, and Zhihua Qu⁵

¹ Meltran, Charlottesville, VA, USA, tom@meltran.com

² Colorado School of Mines, Golden, CO, USA, qiuahuahuang@mines.edu

³ Pacific Northwest National Laboratory, Richland, WA, USA,
yuan.liu@pnnl.gov, meghana.ramesh@pnnl.gov

⁴ Washington State University, Pullman, WA, USA, daniel.glover1@wsu.edu

⁵ University of Central Florida, Orlando, FL, USA,
kenneth.mcdonald@ucf.edu, ganesh.marasini@ucf.edu, qu@ucf.edu

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 renewable resources. Inverter-based resources include solar photovoltaics, battery storage, and many wind power systems. These resources behave differently than conventional generators, and it's essential to model them efficiently and accurately for the reliable grid integration of renewable energy. This paper applies a customized Hammerstein Wiener framework to train generalized block diagram models from thousands of electromagnetic transient simulations or laboratory test records, including customized loss functions to achieve numerical stability. The block diagram models are integrated with larger grid simulations as voltage-controlled current sources or current-controlled voltage sources. 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. The block diagram models run in several simulators and co-simulation frameworks.

Keywords: Deep learning, inverters, photovoltaic systems, power system transients, system identification.

* 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.

1 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). When conventional rotating machine generators (fossil, hydro, nuclear) predominate, the IBRs operate in a grid-following (GFL) mode, in which the grid establishes voltage and frequency at the IBR terminals. As more IBRs connect to the grid, more of them operate in a grid-forming (GFM) mode to establish voltage and frequency in a grid with little or no rotating machine generation. Among other benefits, GFM IBR can provide restoration service after a grid outage. This paper is focused on solar IBR operating as GFM, but the approach also applies to GFL operation, battery energy storage, and wind generation with full-converter interface.

1.1 Electromagnetic Transient Modeling

Rotating machine generators are represented with well-understood equivalent circuit models in a $dq0$ reference frame [1]. Rotating machine models 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, which should improve model portability. IBR vendors only provide such models under strict confidentiality agreements.

1.2 Data-Driven Modeling

Data-driven models offer some advantages over EMT models:

- They can hide proprietary design details of the inverter from the model user.
- Simulation times are much faster.

- The data-driven model behavior is nonlinear, but continuous and differentiable. These properties are helpful to control system design and simulation. Switching EMT models are often discontinuous and non-differentiable.
- Data-driven models can be implemented in standard functions or “blocks” available in the end-user’s EMT simulator.

An EMT simulator produces transient output data, sometimes called “waveforms” or “point-on-wave” data. Laboratory tests of IBR also produce transient output data [3]. A data-driven model could be trained from either source of transient output data, i.e., it can be trained from either EMT or laboratory test data. This means that 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. 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 used the SI Toolbox functions early in this project, but needed to scale up the problem size, customize the loss functions, and train the models faster. We settled on *dynoNet* [10] for the following reasons:

- 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].
- *dynoNet* trained our models several times faster than the SI Toolbox.

We used the HW framework in this project but had to customize it. In this way, our experience was similar to that reported in [8], which customized ANN.

1.3 Contributions

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

- Examples include single-phase and three-phase inverters, trained from EMT and lab data. The EMT data comes from average and switching models.
- Fitting models in the $dq0$ frame mitigates some issues identified in [9].
- Guidelines have been developed for model architecture, hyperparameter selection, input feature selection, data preparation.

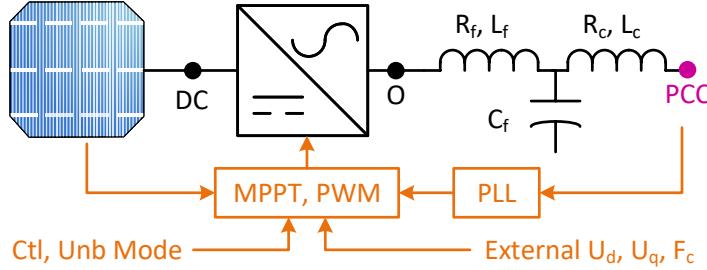


Fig. 1: Functional diagram of solar IBR characteristics in the HWPV model.

- *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 MATLAB/Simulink, the Alternative Transients Program (ATP) [12], Raspberry Pi, a co-simulation framework [13], and a Python test harness.
- Code and examples have been published open-source at [14].

2 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 in Hz. 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.

Reference [15] 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 [16] for EMT simulation, with an incremental conductance MPPT implementation from [17]. In GFL mode, the PLL is

necessary to implement a $dq\theta$ transformation of voltages and currents. In GFM mode, the control frequency, F_c , defines the $dq\theta$ transformation via (1), (2), and (3). A controlled current source (Norton equivalent) will inject phase currents at the PCC according to (4). The controlling phase voltages transform to $dq\theta$ components according to (5). The root mean square (RMS) voltage, used to create a polynomial input feature to the HWPV model, is defined in (6) for three-phase inverters. For single-phase inverters, the factor 1.5 becomes 1.0 in (6). In single-phase inverters, the second order generalized integrator (SOGI) and frequency locked loop (FLL) [18, 19] produce the dq signals.

$$\omega_c = 2\pi F_c \quad (1)$$

$$\theta = \theta_o + \int_{t_0}^t \omega_c dt \quad (2)$$

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

$$I_\phi = T_{dq0}^{-1} [I_d \ I_q \ 0]^T \quad (4)$$

$$V_{rms} = T_{dq0} [V_a \ V_b \ V_c]^T \quad (5)$$

$$V_{rms} = \sqrt{1.5(V_d^2 + V_q^2)} \quad (6)$$

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

- Weather inputs for solar irradiance, G , in W/m^2 , and temperature, T , in C .
- External control inputs U_d , U_q , Ctl , and Unb in pu, and F_c in Hz.
- Transformed voltages, V_d and V_q in V, from V_ϕ at the PCC. Early attempts to use 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 ($Ctl = 0$).

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

- DC voltage, V_{dc} , and current, I_{dc} . These were used for control system inputs in a companion paper [20], but were not connected to a DC circuit.
- AC currents, I_d and I_q in A, transformed to I_ϕ for injection at the PCC.

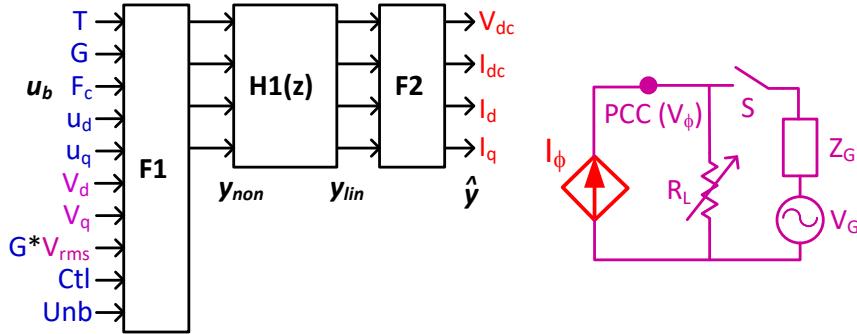


Fig. 2: Block diagram model and controlled current source with grid interface.

The switch, S , is open for GFM and closed for GFL. If the GFM load is purely resistive, as shown in Fig. 2, the dq axes are decoupled, with voltages and currents related as in (7). Otherwise, the dq axes are coupled through the LCL filter and any reactive components of the load and grid impedances.

$$V_d = I_d R_L; V_q = I_q R_L \quad (7)$$

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.

3 Training and Validation

HWPV training begins with collection of hundreds or thousands of transient event records, from either EMT simulation or lab 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{y} , to be < 0.05 .

3.1 EMT and Experimental Systems

Table 1 summarizes the EMT models and hardware specimens used to fit HWPV models. Consortium for Electric Reliability Technology Solutions (CERTS) MicroGrid inverter is three-phase, 480V, 100 kW, and GFM [21]. The *GridLink* is a three-phase, 400V, 5 kW, GFM, software defined inverter (SDI) from Siemens. The *OSG4* is a single-phase, 240V, 13.5 kW inverter derived from an ATP example, enhanced for GFM operation. The *Lab2* specimen is a vendor-supplied single-phase, 240V, 8 kW GFM inverter; it has no access to internal measurement points. N_c is the number of transient records collected for HWPV fitting, and Δt is the EMT simulation time step or instrumentation sample interval.

Table 1: Inverter Systems for Data Collection

Model	Description	Platform	Δt	N_c
CDC	CERTS EMT	PSCAD	20e-6	1
CAC	CERTS EMT	PSCAD	5e-6	1
Bal3	CERTS Average	ATP	10e-6	23,400
OSG4	$1 - \phi$ Switching	ATP	0.5e-6	380
UCF2	GridLink Model	MATLAB	1e-6	1,500
UCF3	GridLink Model	MATLAB	1e-6	2,100
Unb3	CERTS Average	ATP	10e-6	2,430
SDI5	GridLink Hardware	Lab test	5e-5	288
Lab2	$1 - \phi$ GFM	Lab test	0.12e-3	69

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

3.2 Guidelines for Training

Table 2 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* and *UCF3*, 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. Because of bias coefficients in *F1*, the initial conditions on *H1* are not zero even when u_b is zero. In *dynoNet* training, this was accomplished by pre-padding u_b with about 500 points of initial values, then ignoring half the pre-padded region in loss evaluation. In forward evaluation, steady-state initialization of *H1* can be determined from the known initial output of *F1*, denoted y_{non} in Fig. 2, according to (8) applied to each MIMO pair [10].

$$H_1(z = 1) = \frac{b_0 + b_1 + b_2 + b_3 \dots}{1 + a_1 + a_2 + a_3 \dots} \quad (8)$$

Table 3 summarizes the hyperparameters used for each HWPV model:

- Δt is the decimated time step for $H1(z)$
- N_1 is the number of hidden cells in *F1*, each using *tanh* activation. It should be proportional to the number of inputs squared.
- N_2 is the number of hidden cells in *F2*, each using *tanh* activation. It should be proportional to the number of outputs squared.
- N_A is the number of learnable denominator coefficients in $H1(z)$, plus $a_0 = 1$.

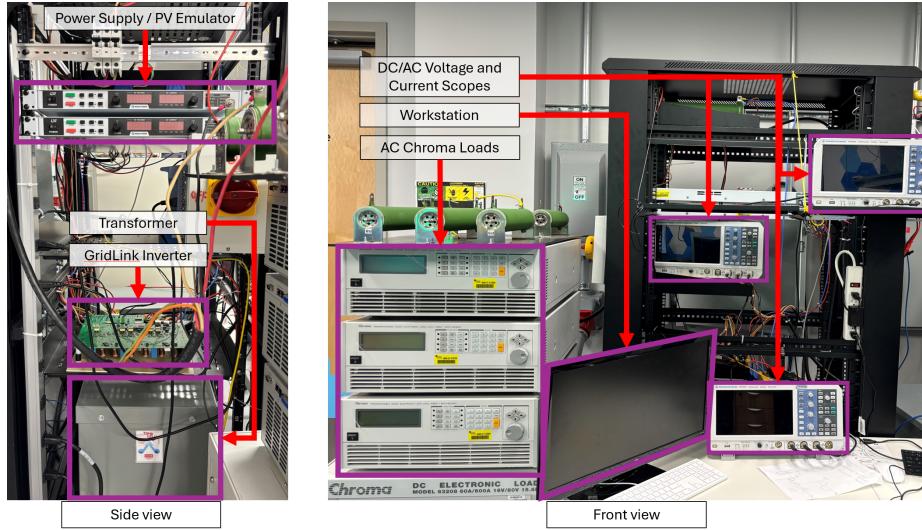


Fig. 3: Testing and data acquisition systems for the GridLink SDI.

- N_B is the number of learnable numerator coefficients in $H1(z)$, 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 the training set size.
- 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 Adam optimizer, used in *PyTorch*.
- N_i is the number of training epochs to meet 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]
- G ramps from 0.0 to one of 10 values in the set [100 to 1000, step 100]
- F_c set at 1 value in [60.0]
- U_d set at one of 2 values in [0.95, 1.0]
- U_q set at one of 3 values in [-0.05, 0.001, 0.05]
- R_L set to achieve one of 3 power P_{pu} values in [0.95, 1.00, 1.05] per (9), where $P_{nom} = 100e3$ and $V_{nom} = 480$

$$R_L = \frac{V_{nom}^2}{P_{nom}} \frac{1000}{G} \frac{1}{P_{pu}} \quad (9)$$

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

Table 2: Inverter Transient Inputs and Outputs

Model	Inputs	Outputs
CDC	G, T	V_{dc}, I_{dc}
CAC	V_{dc}	P_{ac}, Q_{ac}
Bal3	$G, T, F_c, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$	V_{dc}, I_{dc}, I_d, I_q
OSG4	$G, T, F_c, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$	V_{dc}, I_{dc}, I_d, I_q
UCF2	$G, T, F_c, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$	V_{dc}, I_{dc}, I_d, I_q
UCF3	$G, U_d, U_q, Ctl, V_d, V_q, GV_{rms}$	V_{dc}, I_{dc}, I_d, I_q
Unb3	$G, T, F_c, U_d, U_q, Ctl, Unb, V_{dlo}, V_{qlo}, V_{0lo}, V_{dhi}, V_{qhi}, V_{0hi}, GV_{rmslo}$	$V_{dc}, I_{dc}, I_{dlo}, I_{qlo}, I_{0lo}, I_{dhi}, I_{qhi}, I_{0hi}$
SDI5	$F_c, U_d, U_q, V_d, V_q, V_{dc}$	I_{dc}, I_d, I_q
Lab2	F_c, V_c, R_c, V_{dc}	V_{rms}, I_{rms}, I_{dc}

Table 3: HWPV Model Hyperparameters

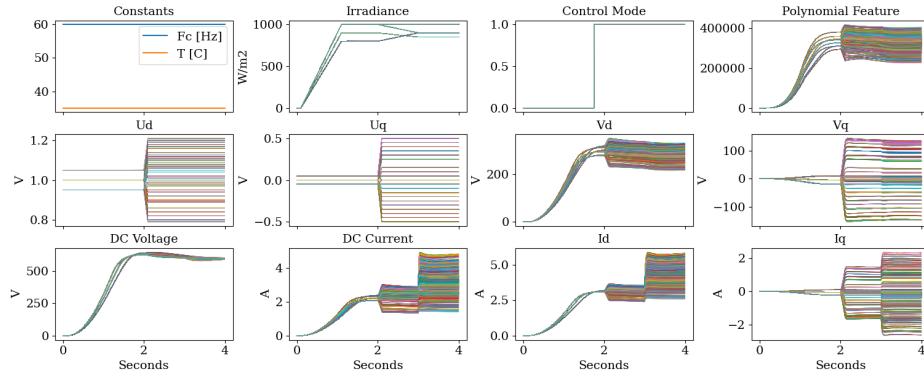
Model	Δt	N_1	N_2	N_A	N_B	N_b	P_v	lr	ϵ	N_i
CDC	0.020	20	20	4	4	1	0	5e-4	1e-8	3500
CAC	0.005	20	20	4	4	1	0	5e-4	1e-8	3500
Bal3	0.002	80	60	2	2	20	10	1e-3	1e-8	1000
OSG4	0.001	80	60	4	4	19	15	5e-4	1e-8	2000
UCF2	0.002	80	60	2	2	50	15	1e-3	1e-8	2000
UCF3	0.001	80	60	2	2	105	15	1e-3	1e-8	1000
Unb3	0.002	120	60	2	2	27	10	1e-3	1e-8	1000
SDI5	0.001	60	40	2	2	18	19	1e-3	1e-6	5000
Lab2	1.2e-4	30	30	2	2	10	10	1e-3	1e-8	1000

- 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 $0 \leq G \leq 1000$
- F_c steps by 1 of 10 values in [-5 to 5, step 1, excluding 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 in [-0.50 to 0.50, step 0.05, excluding 0.0]
- R_L is divided by one of 20 values in [0.80 to 1.20, step 0.02, excluding 1.0]

Fig. 4 shows a different strategy used in collecting 2,100 records for *UCF3*, which is used for external control design and testing in [20]. The T and F_c channels are not used. The ramps in G from 0.1 to 1.1 cover an expected range of startups, while U_d and U_q cover the expected range of initial conditions. At 2.0 s, a change in G , U_d , or U_q occurs. At 3.0 s, a change in R_L occurs.

3.3 Unbalanced Conditions

In (5), the transformed θ components are zero except under unbalanced conditions, when $Unb = 0$. In the field, Unb might come from a zero-sequence

Fig. 4: Sample training set for *UCF3* model with 2100 records.

detection function built into the IBR, or in a local protective relay. The transformation of unbalanced voltages in (5) 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 θ 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 $dq\theta$ current separately as I_{dlo} , I_{qlo} , I_{0lo} , I_{dhi} , I_{qhi} , and I_{0hi} . The total $dq\theta$ currents are constructed from (10), (11), and (12). These $dq\theta$ currents oscillate, but (4) still applies to inject the phase currents at PCC. Fig. 5 shows a sample set of these filtered components used to train the *Unb3* model.

$$I_d = I_{dlo} + I_{dhi} \sin(4\pi F_c t) \quad (10)$$

$$I_q = I_{qlo} + I_{qhi} \sin(4\pi F_c t) \quad (11)$$

$$I_0 = I_{0lo} + I_{0hi} \sin(2\pi F_c t) \quad (12)$$

4 Stability Analysis

It's not enough that the trained HWPV model meet RMSE acceptance criteria; it must also be stable in forward evaluation when connected to a grid.

4.1 Conversion to s-domain

The trained $H1(z)$ implements a fixed Δt as listed in Table 3. 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

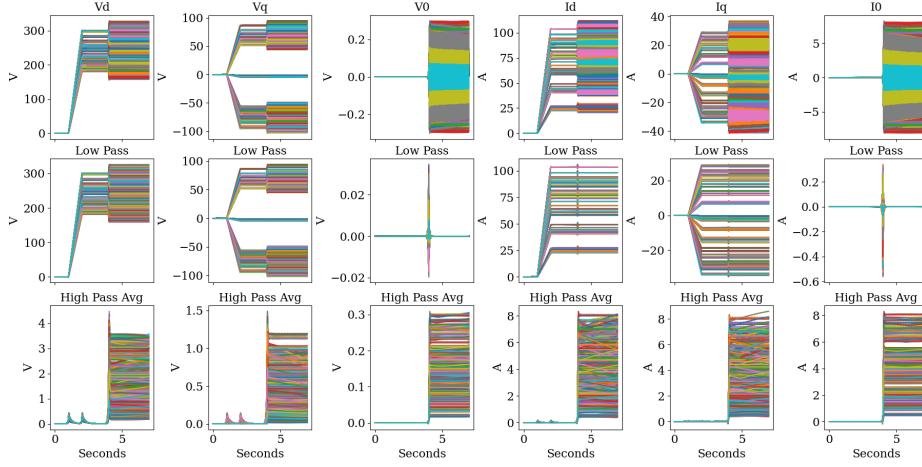


Fig. 5: Filtered unbalanced $dq0$ voltage and current components (2430 records).

used the Python packages *control* [22] and *harold* [23] to convert $H1(z)$ to $H1(s)$. In this conversion, the Forward Euler method in *harold.undiscretize* was found to produce stable poles in $H1(s)$, while Backward Euler and Bilinear Transform sometimes failed to do so. A small number of trained HWPV models with infinite impulse response (IIR) $H1(z)$ produce unstable poles (none reported in the paper), so this must be checked for each IIR model. The *dynoNet* stable second order function option for $H1(z)$, where $N_A = 2$ and $N_B = 3$, has not produced unstable poles in $H1(s)$, which would be an important advantage. To achieve higher orders in $H1(z)$, stable second order blocks might be connected in series.

4.2 Sensitivity

The Norton controlled current source in Fig. 2 is usually compatible with grid system dynamic and EMT solvers, many of which 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 $F1$. This closes a feedback loop, and the constraint (13) must hold for stable operation of the model. In weak grid conditions for the *UCF3* model, this constraint was not satisfied.

$$\sigma = \max\left(\frac{\partial I_d}{\partial V_d}, \frac{\partial I_d}{\partial V_q}, \frac{\partial I_q}{\partial V_d}, \frac{\partial I_q}{\partial V_q}\right) < \frac{1}{R_L} \quad (13)$$

Norton models were retrained by adding violations of (13) to the loss, attempting to optimize σ along with RMSE. Static values of σ were checked over the expected range of input variables, using (8) for efficiency, but the training time still increased substantially. Thevenin models were trained to optimize RMSE and σ by inverting each term in (13) for the loss function.

Table 4: Model Output Fitting Accuracy Results for IIR Norton Models

Model	Normalized Root Mean Square Error (RMSE) ^a								Time [hr] ^b
CDC	V_{dc} 0.0396	I_{cc} 0.0114							
CAC	P_{ac} 0.0049	Q_{ac} 0.0049							
Bal3	V_{dc} 0.0096	I_{dc} 0.0099	I_d 0.0077	I_q 0.0020					28.41
OSG4	0.0223	0.0123	0.0171	0.0057					1.36
UCF2	0.0069	0.0057	0.0060	0.0031					2.26
UCF3 ^c	0.0096	0.0373	0.0478	0.0146					0.35
Unb3	V_{dc} 0.0097	I_{dc} 0.0058	I_{dlo} 0.0078	I_{qlo} 0.0032	I_{0lo} 0.0026	I_{dhi} 0.0048	I_{ghi} 0.0045	I_{0hi} 0.0031	9.29
SDI5	V_{dc} 0.0290	I_{dc} 0.0285	I_d 0.0439						0.77
Lab2	V_{rms} 0.0068	I_{rms} 0.0252	I_{dc} 0.0276						0.06

^aRMSE to be ≤ 0.05 . ^bWith Intel i7-13700K, 64 GB RAM, RTX 4070 Ti card.^cThis model was trained with $\Delta t = 0.01$. Models in Table 5 used $\Delta t = 0.001$.

5 Results and Discussion

5.1 Model Accuracy

Table 4 shows the RMSE acceptance criteria met for all models presented, over all records in the data set. 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 0.0077, it's 0.0561 for this case, and 11 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 201 in the *Bal3* model. The maximum value 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. Table 5 shows that no models met the weak-grid stability criteria for both RMSE and σ , within 1000 epochs.

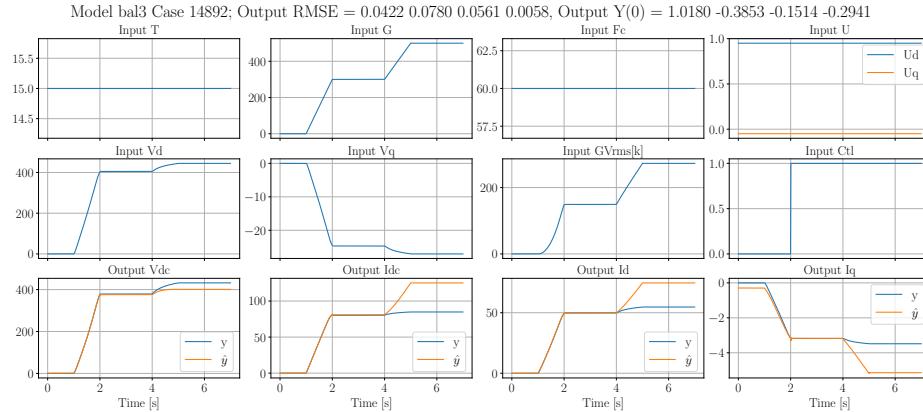
In some control applications, the voltage at node O in Fig. 1 is wanted instead of at the PCC. Given the LCL filter parameters, phasor arithmetic produces the voltages and currents at O from conditions at PCC, but this is a non-causal relationship. Fig. 7 (left) shows a voltage source converter (VSC) voltage at point O in a single-phase switching EMT model. This voltage is not suited for HWPV training. Instead, 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). A causal approach to compensating for the LCL filter is described in [20].

Table 5: Thevenin and Norton Models for *UCF3* with σ Optimization

Source ^a	$H1(z)$	RMSE				σ_{target}	σ_{actual}	Time [hr]
		V_{dc}	I_{dc}	$ I_d, V_d $	$ I_q, V_q $			
Norton	IIR 2nd	0.0060	0.0230	0.0233	0.0080	None	0.4178	3.04
Norton	Stable 2nd	0.0050	0.0208	0.0212	0.0074	None	0.3501	3.23
Norton	IIR 2nd	0.0110	0.0230	0.0468	0.0605	0.0099	0.0082	4.52
Norton	Stable 2nd	0.0049	0.0277	0.0297	0.0085	0.0099	0.0649	4.63
Thevenin ^b	IIR 4th	0.0078	0.0041	0.0076	0.0025	None	236.9	3.49
Thevenin	Stable 2nd	0.0077	0.0046	0.0071	0.0021	None	227.0	3.03
Thevenin	IIR 2nd	0.0522	0.0185	0.0553	0.0093	45.0	53.0	5.67
Thevenin	Stable 2nd	0.0450	0.0184	0.0539	0.0105	45.0	68.7	6.63

^a $N_i = 1000$. $\Delta t = 0.001$. Norton model outputs I_d, I_q and requires $\sigma < 0.0099$.

^b Thevenin model outputs V_d, V_q and requires $\sigma < 55.0$.

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

5.2 Applications

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

- Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS) [13], which could support EMT tools that have a HELICS interface.
- A standalone Python test harness that emulates variable grid impedance.
- Lightweight Raspberry Pi implementation with *numpy*.
- ATP’s Transient Analysis of Control Systems (TACS) feature, which is inefficient because it does not support vector arithmetic, and allows only 1000 local variables. ATP’s MODELS feature was no more efficient.
- A native MATLAB/Simulink implementation.

A DLL implementation is under development to support more EMT tools.

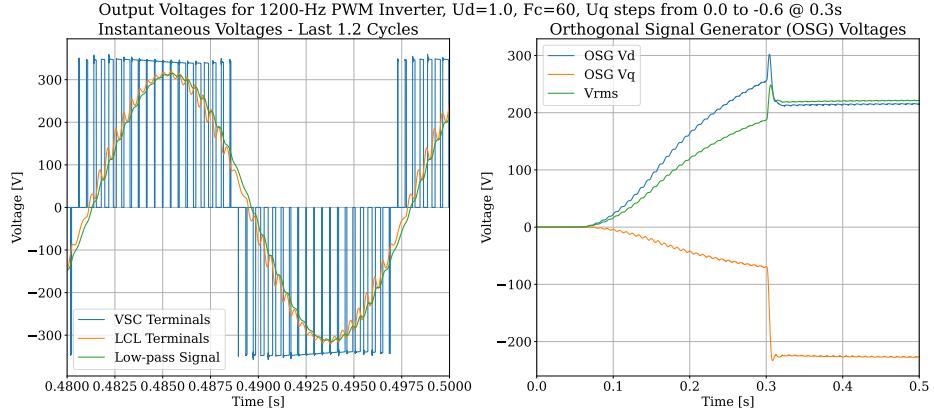


Fig. 7: Voltage outputs from 1-phase PWM inverter model in ATP, system *OSG4*.

6 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. Further work is needed on meeting this constraint, and on coupling Thevenin current-controlled voltage sources to grid system simulations.

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, "Reliability Guideline: Electromagnetic Transient Modeling for BPS Connected Inverter-Based Resources-Recommended Model Requirements and Verification Practices," https://www.nerc.com/comm/RSTC_Reliability_Guidelines/Reliability_Guideline-EMT_Modeling_and_Simulations.pdf, 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 non-linear 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 documentation,” <https://pytorch.org/docs/stable/index.html/>, 2024.
12. European EMTP Users Group, “Alternative Transients Program (ATP-EMTP),” <http://www.emtp.org/>, 2020, [Online; accessed 11-September-2020].
13. 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.
14. T. E. McDermott, “pecblocks documentation,” <https://pecblocks.readthedocs.io/en/latest/>, 2023.
15. 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.
16. 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.
17. 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.
18. 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.
19. R. Teodorescu, M. Liserre, and P. Rodriguez, *Grid Converters for Photovoltaic and Wind Power Systems*. John Wiley & Sons, Ltd, 2011, ch. 4.6, pp. 82–89.
20. G. Marasini, K. McDonald, Z. Qu, and T. E. McDermott, “Data-driven dynamic model and model reference control of inverter based resources,” 2024, submitted to Symposium on Systems Theory in Data and Optimization (SysDO).
21. 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.
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)*. IEEE, 2021, pp. 4875–4881.
23. I. Polat, “An open-source systems and controls toolbox for python3,” <https://github.com/ilayn/harold>, 2018.