

**Document Version**

Final published version

**Licence**

CC BY

**Citation (APA)**

Alsarayreh, S., Dimitrovski, R., & Lekić, A. (2026). DQsym: A MATLAB/Simulink library for dynamic phasor simulation of AC/DC power systems. *SoftwareX*, 34, Article 102719. <https://doi.org/10.1016/j.softx.2026.102719>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

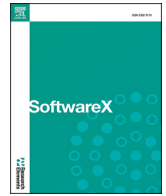
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.  
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

**Sharing and reuse**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.



## *DQ<sub>sym</sub>*: A MATLAB/Simulink library for dynamic phasor simulation of AC/DC power systems

Saif Alsarayreh<sup>a,\*</sup>, Robert Dimitrovski<sup>a,b</sup>, Aleksandra Lekić<sup>a</sup>

<sup>a</sup> Faculty of Electrical Engineering, Mathematics, and Computer Science, TU Delft, Mekelweg 4, 2618 CD, Delft, Netherlands

<sup>b</sup> TenneT TSO GmbH, Bernecker Str. 70, 95448 Bayreuth, Germany

### ARTICLE INFO

#### Keywords:

MATLAB/Simulink library  
Dynamic-phasors  
Power systems  
Simulation  
Converters

### ABSTRACT

AC/DC converter-based power systems require modelling tools that can capture dynamic and harmonic behaviour while integrating seamlessly into established simulation environments. This paper presents *DQ<sub>sym</sub>*, an open-source MATLAB/Simulink library for dynamic phasor-based simulation using a unified state-space formulation. Using *DQ<sub>sym</sub>*, multiple harmonic orders are represented through multiple rotating DQ reference frames within a single simulation run. *DQ<sub>sym</sub>* is implemented using standard MATLAB/Simulink functions and masked subsystems, allowing seamless integration into existing workflows. Available components include sources, mathematical operation blocks, state-space blocks, and measurement tools. *DQ<sub>sym</sub>* provides structured numerical outputs and visualization utilities to support reproducible simulation studies. The design and architecture of *DQ<sub>sym</sub>* are described, and an illustrative example based on a single-station Modular Multilevel Converter (MMC) with full control demonstrates its correctness. Simulation results closely match those obtained from conventional Simulink implementations, validating the approach and confirming its suitability for harmonic-capture power system simulation.

### Metadata

#### Code metadata.

Current code version	v1.0
Permanent link to code / repository	<a href="https://github.com/control-protection-grids-tudelft/DP">https://github.com/control-protection-grids-tudelft/DP</a>
Code Ocean compute capsule	Not available
Legal code license	MIT License
Code versioning system used	Git
Software languages, tools, and services	MATLAB (tested with R2024a), Simulink
Compilation requirements, operating environments & dependencies	<b>1-Core requirements:</b> MATLAB R2024a or newer, Simulink. <b>2-Electrical system modelling:</b> Simscape, Simscape Electrical (formerly SimPowerSystems). <b>3-Optional (used in specific examples and post-processing):</b> Signal Processing Toolbox, DSP System Toolbox. <b>4-Operating system:</b> Tested on Windows 11. No platform-specific dependencies are used; compatibility with Linux and macOS is expected but has not been extensively validated.
Link to developer documentation/manual	<a href="https://github.com/control-protection-grids-tudelft/DP/tree/main/DP_v1.0/Documentation">https://github.com/control-protection-grids-tudelft/DP/tree/main/DP_v1.0/Documentation</a>
Support email for questions	<a href="mailto:S.T.S.Alsarayreh@tudelft.nl">S.T.S.Alsarayreh@tudelft.nl</a>

### 1. Motivation and significance

AC/DC converter-based power grids have become a cornerstone of modern power systems. This development is driven by the widespread

deployment of voltage source converter (VSC) technologies and power-electronic interfaced resources. While these systems offer enhanced controllability and flexibility, they also introduce harmonic interactions and converter-driven dynamics. These effects challenge

\* Corresponding author.

Email address: [S.T.S.Alsarayreh@tudelft.nl](mailto:S.T.S.Alsarayreh@tudelft.nl) (S. Alsarayreh).

conventional modelling approaches because of the fast dynamics and complex behaviour. Accurate system-level analysis, therefore, requires models capable of capturing both fundamental-frequency behaviour and harmonic interactions [1–3].

Electromagnetic transient (EMT) simulation tools provide detailed representations of switching behaviour and associated harmonics. However, their high computational cost limits their applicability to large-scale networks, and they are typically used for small systems or short-duration studies [4,5]. In contrast, classical phasor-domain and average-value models are computationally efficient and well-suited for steady-state analysis. Their main limitation is the neglect of harmonic coupling, which becomes critical in converter-dominated systems [6,7].

Dynamic phasor (DP) theory provides an intermediate modelling approach. It represents time-periodic signals using time-varying Fourier coefficients [2]. Selected harmonic dynamics are retained within a structured representation. DP models, therefore, capture both fundamental and harmonic behaviour within a controllable frequency range and provide an effective balance between modelling detail and analytical tractability [2,8].

Existing work on DP modelling is distributed across different applications and environments, making it difficult to assess and compare capabilities consistently [8]. In particular, standardised and cohesive simulation environments are still lacking.

Code-centric tools such as DPsim [9] provide efficient implementations with support for real-time simulation and co-simulation, but lack support for HVDC converters. In parallel, DP formulations have been explored in equation-based environments such as Modelica, where prior work and open-source libraries (e.g., ModPowerSystems) demonstrate the integration of DP models within that framework [10,11].

These approaches have distinct but related limitations. Code-based tools often require advanced programming expertise, which hinders rapid adaptation. Equation-based environments offer flexibility but involve a steep learning curve and lack seamless integration with control design pipelines. They also do not natively support the state-space

formulations required for stability analysis. Consequently, these tools remain difficult to adopt, particularly for users accustomed to block-based design.

The growing complexity of AC/DC systems highlights the need for modelling tools that are accurate, transparent, and easy to use. However, support for DP-based analysis remains limited. Only a few platforms provide native DP representations. DPsim, for example, includes harmonic-indexed dynamic phasor states [9], while widely used environments such as OpenIPSL (OpenModelica) and PowerSimulationsDynamics.jl do not provide native DP abstractions [11–13]. In these tools, DP models must be implemented manually by introducing additional state variables and defining cross-frequency coupling. Consequently, the DP approach is typically treated as an extension to existing frameworks rather than as a native capability.

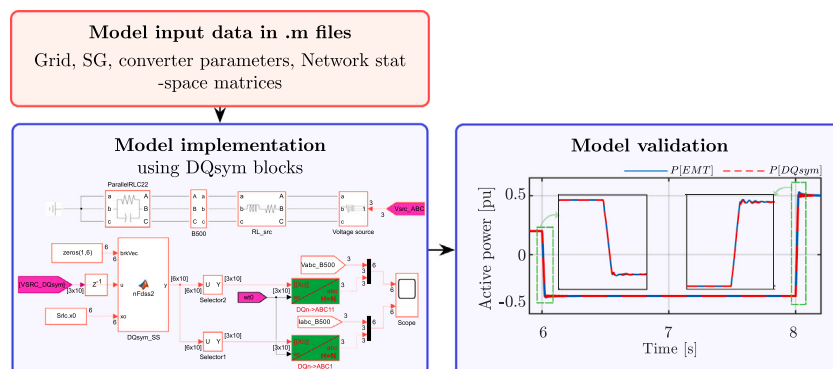
What is therefore needed is a block-based, accessible DP simulation environment that integrates naturally with control design. To address these challenges, this work introduces *DQsym*, an open-source Simulink library for AC/DC power system simulation based on dynamic phasor modelling. Unlike existing implementations, *DQsym* adopts a fully block-based approach. Model construction is intuitive and integrates directly with control design workflows. The library provides modular representations of AC networks, DC networks, converters, and control systems.

Implementation within the MATLAB/Simulink environment allows users to work within familiar environments. Its modular, case-based architecture supports reproducibility and model reuse, reducing implementation complexity and improving accessibility for system-level studies (see Table 1).

The significance of *DQsym*, illustrated in Fig. 1, lies in its ability to bridge the gap between EMT and conventional phasor-based approaches by enabling harmonic-rich modelling within a unified state-space framework. Rather than focusing on reducing computational cost relative to EMT simulations, capturing system dynamics across a wide range of harmonic components is the primary design objective. The same

**Table 1**  
Open-source tools for power system dynamic simulation.

Tool	Environment	DP support / modelling paradigm	Capabilities	Limitations / Notes
<i>DQsym</i>	MATLAB Simulink	Explicit DP (state-space, block-based)	System-level AC/DC studies, reproducible workflows, harmonic modelling	Simulink dependency, scalability tied to model structure
DPsim [9]	C++ / FMI	Explicit DP + EMT (hybrid)	Real-time simulation, HIL, co-simulation, high fidelity	Higher implementation complexity
PowerSimulations Dynamics.jl [12]	Julia	RMS / phasor-based (no native DP)	Large-scale system studies, flexible dynamic modelling	Limited harmonic representation
PowerDynamics.jl [14]	Julia	DAE-based (phasor, no native DP)	Flexible equation-based modelling, stability studies	No explicit DP support
OpenModelica libraries [10,11,13]	Modelica	DAE-based (extensible to DP)	Component-level modelling, multi-domain simulation, object-oriented design	Requires manual DP implementation, higher modelling complexity



**Fig. 1.** Workflow of executing a simulation using the *DQsym* library in Matlab/Simulink.

model can therefore be applied to both time-domain simulation and frequency-domain stability analysis without reformulation.

Simulation outputs are available as time-domain signals, supporting flexible analysis of system behaviour. Multi-harmonic state-space formulation underpins the framework and provides a foundation for future extensions toward stability analysis and control design.

The remainder of this paper is organised as follows. Section 2 introduces the  $DQ_{sym}$  library, detailing its Simulink architecture, modelling philosophy, and key functionalities. Section 3 presents simulation results from a representative test case. Section 4 discusses the implications and potential impact of  $DQ_{sym}$  for AC/DC system analysis. Finally, Section 5 concludes the paper and outlines directions for future work.

## 2. Software description

### 2.1. Software architecture

$DQ_{sym}$  is implemented as a Simulink library for building power system models in a structured way. The library is organised into groups of blocks, including AC and DC sources, power electronic converters (currently a modular multilevel converter implemented using a state-space model), mathematical and dynamic equation blocks, and measurement and post-processing blocks. Each group is implemented as a masked subsystem. This separation ensures a clear modelling structure and consistent interfaces between components.

Each component is implemented as a parameterised subsystem that contains the dynamic phasor equations. All blocks share a common input–output structure and use a unified  $DQ_{sym}$  signal format, defined in Eq. (3) [15]. The modular structure allows components to be added, replaced, or extended without modifying the overall system layout, supporting reuse across different study cases.

System models follow the standard Simulink workflow. Users define parameters and simulation settings either in MATLAB scripts or directly in block masks. These values configure the selected library blocks. Users place and connect blocks to form the desired network, then execute simulations using standard Simulink solvers without modifying internal block implementations. During simulation, results can be visualised using MATLAB built-in tools.

### 2.2. Dynamic phasor representation

The  $DQ_{sym}$  framework is based on DP theory, where time-varying signals are represented using Fourier coefficients that evolve over time. For a nearly periodic signal  $x(t)$ , the dynamic phasor of order  $k$  is defined as

$$\langle x \rangle_k(t) = \frac{1}{T} \int_{t-T}^t x(\tau) e^{-jk\omega_c \tau} d\tau, \quad (1)$$

where  $\omega_c = 2\pi/T$  is the fundamental frequency. Each coefficient  $\langle x \rangle_k(t)$  captures the amplitude and phase of the  $k^{\text{th}}$  harmonic component.

Using this representation, a signal can be approximated as a sum of selected harmonic components:

$$x(t) \approx \sum_{k=-N}^N \langle x \rangle_k(t) e^{jk\omega_c t}, \quad (2)$$

where  $N$  is the selected harmonic order. Increasing  $N$  improves accuracy but also increases computational effort.

Dynamic phasors provide a compact representation of harmonic-rich signals while preserving time-domain dynamics. Their algebraic properties allow the direct formulation of system equations. In particular, multiplication introduces coupling between harmonic components through the convolution of harmonic indices.

In the  $DQ_{sym}$  framework, dynamic phasors are expressed in a rotating  $dq0$  reference frame and organised into positive-, negative-, and zero-sequence components. Each signal is represented as a set of complex-valued coefficients obtained by mapping from the  $abc$  frame

to the  $DQ_{sym}$  frame through a symmetrical component transformation followed by a rotation into a synchronous reference frame. This process separates positive-, negative-, and zero-sequence components while retaining the amplitude and phase of each harmonic component. A detailed derivation is provided in [15,16]. These coefficients form the standard signal structure used across all library blocks, as defined in Eq. (3).

$$\begin{bmatrix} \langle x_{dq,p} \rangle_k \\ \langle x_{dq,n} \rangle_k \\ \langle x_{dq,z} \rangle_k \end{bmatrix} = \begin{bmatrix} \mathbf{x}_d^p(k) + j\mathbf{x}_q^p(k) \\ \mathbf{x}_d^n(k) + j\mathbf{x}_q^n(k) \\ \mathbf{x}_d^z(k) + j\mathbf{x}_q^z(k) \end{bmatrix} \quad (3)$$

#### 2.2.1. Simulink block library - library components

All components described in this section are available in the  $DQ_{sym}$  library (version 1.0). They are implemented as masked subsystems that contain their internal algorithms. The algorithms presented here provide a high-level description of each block's operation, not the complete implementation. The detailed implementation can be inspected within the masked blocks after installation. Each algorithm is associated with a specific block in the library. Algorithm 1 corresponds to the state-space block, Algorithm 2 corresponds to the multiplication block, and Algorithm 3 corresponds to the  $DQ_{sym}$ -to-ABC transformation block. Once installed,  $DQ_{sym}$  appears as a standard Simulink library in the Simulink Library Browser, where users interact directly with the masked blocks.

**Source and grid interface blocks.** Source and grid interface blocks provide voltage or current inputs in the  $DQ_{sym}$  dynamic phasor format defined in (3) [15]. These blocks represent ideal AC or DC sources. Signals are defined through block masks and expressed in multiple DQ reference frames based on the selected harmonic order  $N$ . For visualisation, signals can be converted to the  $abc$  reference frame and displayed using standard Simulink scope blocks. The blocks are fully parameterised and do not require modification of their internal implementation.

---

**Algorithm 1** Time-stepping with Switching-Event Handling and Half-step Correction.

---

- 1: **Get system matrices** (continuous-time or discrete-time representation)
- 2: **if Discrete system matrices available? then**
- 3:   Set  $(A, B, C, D)$
- 4: **else**
- 5:   **Discretize system matrices** to obtain  $(A_d, B_d, C_d, D_d)$
- 6: **end if**
- 7: **for** each time-step  $k = 0, 1, 2, \dots$  **do**
- 8:   **Half-step correction**
- 9:   Apply predictor/corrector (or midpoint) update to align states/inputs at  $t_{k+\frac{1}{2}}$
- 10:   **if Switching event occurs during**  $[t_k, t_{k+1}]$  **then**
- 11:     **Update system matrices**
- 12:     Recompute (or select)  $(A_d, B_d, C_d, D_d)$  for the new switch configuration
- 13:     **Half-step correction**
- 14:     Re-apply half-step correction with updated matrices
- 15:   **end if**
- 16:   **Solve system state-space formulation**
- 17:   Propagate state and outputs, e.g.

$$x[k+1] = A_d x[k] + B_d u[k], \quad y[k] = C_d x[k] + D_d u[k]$$

- 18:   **Outputs:** report/store  $y[k]$
  - 19: **end for**
-

**Algorithm 2**  $DQ_{sym}$  Multiplication Block.

**Require:** Input signals  $x, y$  in  $DQ_{sym}$  representation (up to harmonic order  $n$ )

**Ensure:** Output signal  $z$  in  $DQ_{sym}$  representation

- 1: **Step 1: Transformation to phase domain**
- 2: Convert  $x$  and  $y$  from  $DQ_{sym}$  to  $ABC$  coordinates

- 3: **Step 2: Harmonic decomposition**

- 4: **for**  $k = 0$  to  $n$  **do**

- 5:     Extract cosine and sine Fourier coefficients:

$$x_k = (a_k^x, b_k^x), \quad y_k = (a_k^y, b_k^y)$$

- 6: **end for**

- 7: **Step 3: Coefficient multiplication**

- 8: **for**  $k = 0$  to  $n$  **do**

- 9:     Compute product coefficients:

$$a_k^z = a_k^x a_k^y - b_k^x b_k^y$$

$$b_k^z = a_k^x b_k^y + b_k^x a_k^y$$

- 10: **end for**

- 11: **Step 4: Back transformation**

- 12: Reconstruct  $z$  in  $ABC$  domain from  $\{a_k^z, b_k^z\}$

- 13: Transform result from  $ABC$  back to  $DQ_{sym}$

- 14: **return**  $z$

**Algorithm 3** Transformation from  $DQ_{sym}$  to  $ABC$  (up to harmonic order  $n$ ).

**Require:** Input signal in  $DQ_{sym}$  format up to harmonic order  $n$ :

$$X_{dq}^{p,n,z} = \begin{bmatrix} x^p(0) & \dots & x^p(n) \\ x^n(0) & \dots & x^n(n) \\ x^z(0) & \dots & x^z(n) \end{bmatrix}$$

**Ensure:** Output signal  $x_{abc}$  in  $ABC$  format

- 1: **Transformation** ( $DQ_{sym} \rightarrow ABC$ )

- 2: Compute

$$x_{abc} \leftarrow T_{pnz}^{-1} \left\{ X_{dq}^{p,n,z} \right\}$$

- 3: **return**  $x_{abc}$

**Converter and power electronics blocks.** Power electronic converters are currently represented by a modular multilevel converter (MMC). The model is implemented using a discrete-time representation. Electrical ratings, control parameters, and operating conditions are defined through block parameters. Inputs include DQ-frame voltages, currents, and control signals. Outputs are computed using the system matrices. This formulation can be extended to other averaged converter models.

**State-space block.** The state-space block provides a general way to implement dynamic component models. It can represent converters, network elements, and control systems described by differential equations.

Block inputs include  $DQ_{sym}$ -formatted electrical variables, control signals, and disturbances. Outputs include state variables and computed electrical quantities. Internally, the model uses a discrete-time state-space formulation.

The block also handles switching events by updating system matrices mid-step, as described in [Algorithm 1](#). Switching behaviour is thereby modelled consistently.

**Mathematical operations blocks.** Mathematical operation blocks provide the numerical functions required for dynamic phasor modelling, including matrix multiplication, integration, and harmonic interaction calculations. All operations preserve the  $DQ_{sym}$  signal format throughout. [Algorithm 2](#) presents the multiplication process.

**Measurement and post-processing blocks.** Post-processing blocks convert  $DQ_{sym}$  signals into time-domain waveforms. These can be visualised, logged, or compared with EMT simulations. [Algorithm 3](#) describes the transformation to the  $abc$  frame. Measurement blocks, such as Mux and Demux, support signal routing and integration with standard Simulink components.

### 2.2.2. Installation, simulation execution and result handling

$DQ_{sym}$  provides a unified simulation and data handling procedure. Simulations are executed using standard Simulink solvers, and selected signals are logged during runtime. After simulation, results can be transformed to the  $abc$  reference frame and stored as structured MATLAB variables.

The framework integrates with MATLAB/Simulink visualisation tools, enabling intuitive analysis of system behaviour. Post-processing can be performed using standard MATLAB functions, including FFT-based analysis. Installation is performed by running the MATLAB script `addlib.m`, located in the  $DQ_{sym}$  library folder, which adds the toolbox to the MATLAB path and registers the library in the Simulink Library Browser.

The modelling approach assumes that users can obtain a discrete-time state-space representation of the system. Version 1.0 includes an averaged MMC model. More detailed converter models require additional development. Generator models with mechanical dynamics are not currently supported.

The main goal of  $DQ_{sym}$  is not to reduce simulation time compared to electromagnetic transient (EMT) models. Although this approach may increase computational effort compared to simplified models, it provides improved modelling accuracy and analytical flexibility. The main advantage of  $DQ_{sym}$  lies in linking harmonic modelling and system-level stability analysis within a single framework.

### 2.2.3. Dependencies and environment

The  $DQ_{sym}$  library is developed and tested using MATLAB R2024a. Core functionality relies on Simulink and Simscape Electrical. Additional toolboxes, including the Signal Processing Toolbox and DSP System Toolbox, are required for specific functions. The library has been tested on Windows 11. No platform-specific features are used. Compatibility with Linux and macOS is expected, although it has not been formally validated.

## 3. Illustrative examples

$DQ_{sym}$  library includes several example models for demonstration and validation. The provided examples comprise a simple RLC system, a point-to-point HVDC system, and an IEEE 9-bus system extended with a point-to-point HVDC link to form a hybrid AC/DC network. Additional implementation details, including step-by-step model construction and block-level descriptions, are available in the user manual provided with the open-source repository.

### 3.1. Multiplication of harmonic-rich signals

This example illustrates the multiplication of two harmonic-rich signals, each containing components up to the 5th harmonic order. An additional independent 10th-order component is added, as defined

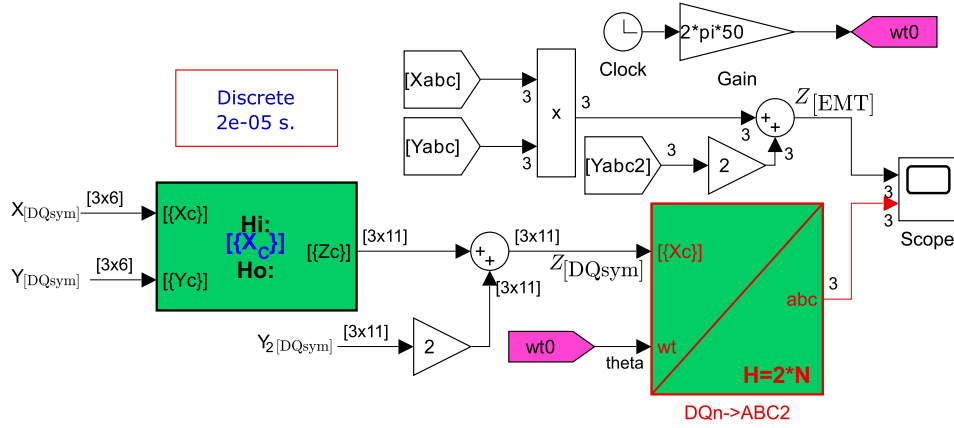


Fig. 2. Overview of the multiplication and measurement blocks in both the SPS and the  $DQ_{sym}$  libraries.

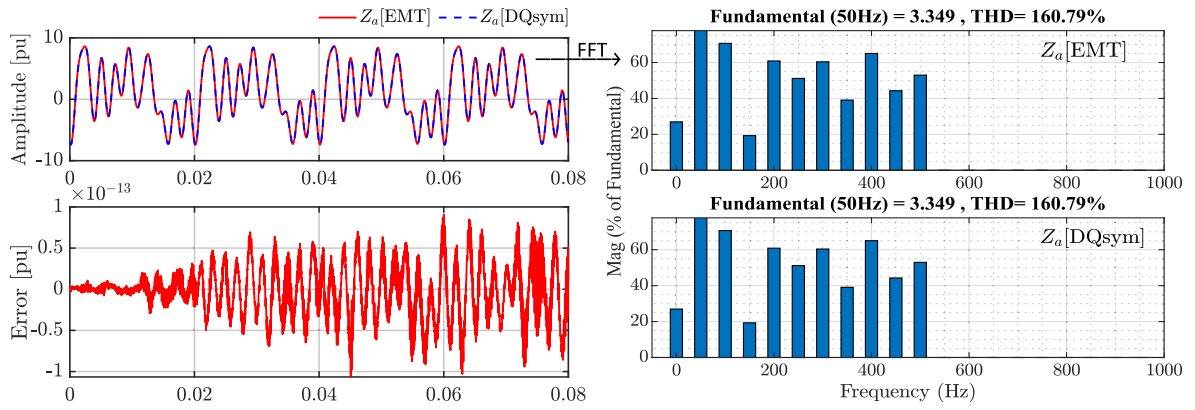


Fig. 3. Multiplication block validation using the  $DQ_{sym}$  library. Time-domain comparison with EMT results and FFT-based spectral Output confirms consistency in harmonic content and THD values.

in (4). The implementation using  $DQ_{sym}$  blocks and the corresponding EMT-based MATLAB/Simulink reference model is shown in Fig. 2. This example serves as a minimal validation case.

The multiplication stage introduces harmonic coupling terms consistent with sum- and difference-frequency interactions ( $|h_1 \pm h_2|$ ). The time-domain waveforms obtained from  $DQ_{sym}$  closely match the EMT reference, as shown in Fig. 3, confirming correct harmonic indexing and alignment.

FFT spectral analysis shows full agreement between both approaches across all measured frequency components, including cross-coupled terms. The 50 Hz fundamental magnitude (3.349) and total harmonic distortion (THD = 160.79%) are identical in both cases, confirming the numerical consistency of the multiplication and measurement blocks and the fidelity of the dynamic phasor framework in representing nonlinear harmonic interactions.

$$\mathbf{Z}_{DQsym} = \mathbf{X}_{DQsym} \times \mathbf{Y}_{DQsym} + 2 \times \mathbf{Y}_{2DQsym}$$

$$= \begin{pmatrix} 0 & -0.88 + 0.40i & -0.01 + 0.00i & -0.12 - 0.12i & -0.34 - 0.13i & -0.89 - 0.09i \\ 0 & 0.04 - 0.69i & -0.11 - 0.65i & 0.47 + 0.87i & -0.66 + 0.34i & 0.40 + 0.58i \\ 0 & -0.24 - 0.36i & 0.41 + 0.34i & -0.21 + 0.16i & 0.30 + 0.30i & -0.92 - 0.24i \end{pmatrix} \times \begin{pmatrix} 0 & 0.08 - 0.25i & 0.06 + 0.08i & -0.80 + 0.57i & 0.63 - 0.17i & 0.55 + 0.51i \\ 0 & 0.06 - 0.48i & 0.06 + 0.16i & 0.35 + 0.51i & -0.90 + 0.24i & -0.24 + 0.47i \\ 0 & -0.06 + 0.39i & -0.13 + 0.29i & 0.20 + 0.32i & 0.02 - 0.24i & -0.44 + 0.32i \end{pmatrix} + 2 \times \begin{pmatrix} 0 & 0.6 + 0.1i & 0.7 - 0.5i & -0.5 - 0.3i & -0.1 - 0.2i & 0.4 + 0.1i & -0.2 - 0.2i & \dots \\ 0 & 0.3 - 0.9i & 0.4 + 0.5i & 0.4 + 0.3i & -0.3 - 0.3i & -0.5 + 0.2i & -0.4i & \dots \\ 0 & 0.3 - 0.7i & 0.1 + 0.2i & 0.7 & 0.1 + 0.2i & 0.4 - 0.9i & -0.4 & \dots \end{pmatrix} \quad (4)$$

**Table 2**  
MMC Station Data.

Parameter & Description	Value
$S_b$ : Base power	1000 MVA
$R_{arm}$ : Arm resistance	166 m $\Omega$
$R_G$ : Equivalent transformer resistance	400 m $\Omega$
$R_L$ : Equivalent load resistance	240 $\Omega$
$L_{arm}$ : Arm inductance	52.94 mH
$L_G$ : Equivalent transformer inductance	52 mH
$C$ : Submodule capacitance	1.758 mF
$N$ : Number of submodules	216
$\omega_g$ : Fundamental grid frequency	$2\pi \cdot 50$ rad/s
$V_{DC}$ : Rated DC voltage	640 kV
$V_{PCC}$ : PCC voltage	400 kV

### 3.2. MMC station

A single MMC station serves as the reference system for software validation. A model is developed in MATLAB/Simulink using the  $DQ_{sym}$  library and benchmarked against a high-fidelity reference implementation. The MMC is modelled in state-space form, with arm capacitor voltage dynamics derived from Kirchhoff's current law and expressed in terms of the circulating current, grid current, and modulation signals:

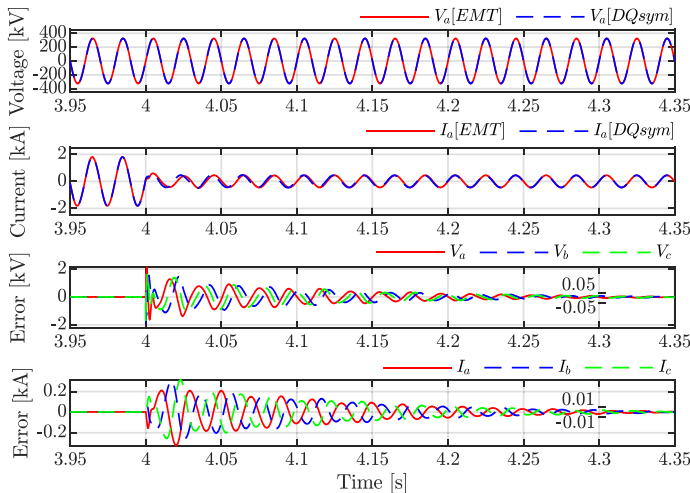
$$\frac{du_{cp}^{\Sigma}}{dt} = \left( i_c + \frac{i_g}{2} \right) \frac{m_p}{C_{arm}}, \quad \frac{du_{cn}^{\Sigma}}{dt} = \left( i_c - \frac{i_g}{2} \right) \frac{m_n}{C_{arm}}.$$

All state variables are expressed in dynamic phasor form within the  $DQ_{sym}$  framework, retaining fundamental-frequency and harmonic components in a unified formulation. The controller encompasses active and reactive power regulation and circulating current suppression, with all signals processed directly in the  $DQ_{sym}$  domain. The station parameters are listed in Table 2.

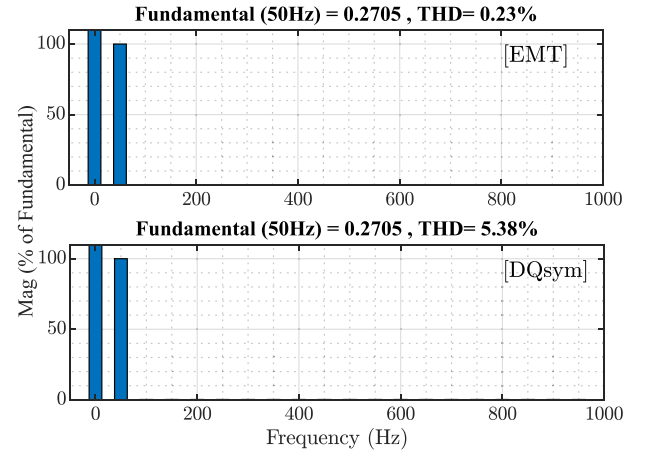
#### 3.2.1. Comparison of $DQ_{sym}$ and EMT simulations

Figs. 4 and 5 compare EMT and  $DQ_{sym}$  results for voltage, current, arm currents, and harmonic spectra. Results are transformed to the  $abc$  frame for comparison.

Phase- $a$  PCC voltage and grid current show close agreement between both models. The maximum voltage deviation is approximately 50 V, corresponding to 0.015% of the nominal value. The maximum current deviation is 10 A (0.69%), indicating strong consistency. Following a step change in active power, both models exhibit similar transient responses and quickly reach steady state. Arm current comparisons



**Fig. 4.** Comparison of EMT and  $DQ_{sym}$  simulations for PCC voltage, grid current, and arm currents during step changes.



**Fig. 5.** Frequency-domain comparison of harmonic spectra obtained from EMT and  $DQ_{sym}$  simulations.

also show strong agreement, with transient deviations of approximately  $\pm 0.16$  kA and negligible steady-state error.

Frequency-domain analysis confirms matching spectra across all components. The fundamental is identical between both models, while small THD differences arise from truncation at the fifth order in the  $DQ_{sym}$  formulation—a deliberate modelling choice, not an inherent limitation of the framework.

## 4. Impact

$DQ_{sym}$  provides a practical, block-based modelling approach for AC/DC power systems, combining harmonic-accurate simulation with direct integration into control design workflows. Implemented as a structured library in MATLAB/Simulink, it enables system-level studies that capture dynamic behaviour and harmonic components while remaining compatible with standard engineering tools and workflows.

Intended users include those involved in the planning, operation, and control design of converter-based power systems, including transmission system operators (TSOs) and industrial practitioners.  $DQ_{sym}$  supports scalable studies of converter interactions, harmonic propagation, and control performance in interconnected AC/DC networks. Clear algebraic rules for dynamic phasor operations—such as multiplication, integration, and summation—are formulated to be consistent with

the state-space structure. Its Simulink-based implementation enables integration with existing MATLAB/Simulink environments, supporting design evaluation, parameter studies, and controller testing within existing environments.

As an extensible library, *DQ<sub>sym</sub>* promotes transparent and reproducible model development. It therefore serves as a practical tool for both academic research and industrial applications, supporting the increasing complexity of modern power systems with a growing share of power-electronic converters.

## 5. Conclusion and future work

This paper presents a validated MATLAB/Simulink library for dynamic phasor-based simulation of power systems while preserving a state-space formulation. The library is implemented entirely within the MATLAB/Simulink environment and is designed for seamless integration into established Simulink modelling workflows. It provides a collection of reusable masked components implemented using MATLAB Function blocks and compatible with Simulink elements. The available components include source models, mathematical operation blocks, state-space formulation blocks, and measurement and post-processing blocks.

A clear, structured description of each block is provided, detailing both its underlying implementation and its functional role within the overall modelling framework. A key contribution is the standardised input–output interface shared across all blocks, which supports multiple rotating DQ reference frames. This feature enables the simultaneous representation of multiple harmonic orders within a single simulation run, significantly enhancing the capability to analyze spectral coupling in converter-dominated networks. The state-space formulation directly supports frequency-domain stability analysis, requiring no model reformulation.

To demonstrate the functionality and modelling principles of the *DQ<sub>sym</sub>* library, two illustrative examples are presented: a simple multiplication test case and a single-station MMC with full control. These examples demonstrate the modelling accuracy and harmonic-capture capability of the *DQ<sub>sym</sub>* formulation. Simulation results obtained with the *DQ<sub>sym</sub>* library closely match those of benchmark models constructed with conventional MATLAB/Simulink blocks, yielding overlapping time-domain responses.

## CRedit authorship contribution statement

**Saif Alsarayreh:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Robert Dimitrovski:** Writing – review & editing, Supervision, Software, Resources, Project administration, Methodology, Funding acquisition, Conceptualization. **Aleksandra Lekić:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was funded by the CRESYM project Harmony (<https://cresym.eu/harmony/>).

## References

- [1] Lara JD, Henriquez-Auba R, Ramasubramanian D, Dhople S, Callaway DS, Sanders S. Revisiting power systems time-domain simulation methods and models. *IEEE Trans Power Syst* 2024;39(2):2421–37. <https://doi.org/10.1109/TPWRS.2023.3303291>
- [2] Shah C, Vasquez-Plaza JD, Campo-Ossa DD, Patarroyo-Montenegro JF, Guruwacharya N, Bhujel N, Trevizan RD, Rengifo FA, Shirazi M, Tonkoski R, Wies R, Hansen TM, Cicilio P. Review of dynamic and transient modeling of power electronic converters for converter dominated power systems. *IEEE Access* 2021;9:82094–117. <https://doi.org/10.1109/ACCESS.2021.3086420>
- [3] Taghvaie A, Warnakulasuriya T, Kumar D, Zare F, Sharma R, Vilathgamuwa DM. A comprehensive review of harmonic issues and estimation techniques in power system networks based on traditional and artificial Intelligence/Machine learning. *IEEE Access* 2023;11:31417–42. <https://doi.org/10.1109/ACCESS.2023.3260768>
- [4] North American Electric Reliability Corporation. EMT analysis in power system planning and operations. Tech. Rep. Version 2.0 NERC, Atlanta, GA, USA; Jun 2025.
- [5] Allabadi A, Mahseredjian J, Denetière S, Abusalah A, Kocar I, Ould-Bachir T. Acceleration strategies for EMT simulation of HVDC systems. *Electr Power Syst Res* 2026;252:112399. <https://doi.org/10.1016/j.epsr.2025.112399>. <https://www.sciencedirect.com/science/article/pii/S0378779625009861>.
- [6] Cao Y, Gu W, Xu M, Zhang F, Liu W, Zhang S. Harmonic-preserved average-value model for converters in electromagnetic transient simulation. *IEEE Trans Power Deliv* 2025;1–15. <https://doi.org/10.1109/TPWRD.2025.3645046>
- [7] Becker JKM, Zuo Y, Paolone M, Kettner AM. Harmonic power-flow study of hybrid AC/DC grids with converter-interfaced distributed energy resources. In: 2023 IEEE Belgrade PowerTech; 2023. p. 1–6. <https://doi.org/10.1109/PowerTech55446.2023.10202734>
- [8] Ghimire P, Poudel S, Shirazi M, Hummels DM, Tonkoski R. Application of dynamic phasor modeling in power systems: a review. *IEEE Access* 2026;14:28573–94. <https://doi.org/10.1109/ACCESS.2026.3666150>
- [9] Mirz M, Dinkelbach J, Monti A. DPsim—Advancements in power electronics modelling using shifted frequency analysis and in real-time simulation capability by parallelization. *Energies* 2020;13(15). <https://doi.org/10.3390/en13153879>. <https://www.mdpi.com/1996-1073/13/15/3879>.
- [10] Vanfretti L, Rabuzin T, Baudette M, Murad M. iTesla power systems library (iPSL): a modelica library for phasor time-domain simulations. *SoftwareX* 2016;5:84–8. <https://doi.org/10.1016/j.softx.2016.05.001>. <https://www.sciencedirect.com/science/article/pii/S2352711016300097>.
- [11] Baudette M, Castro M, Rabuzin T, Lavenius J, Bogodorova T, Vanfretti L. OpenIPSL: open-instance power system library — update 1.5 to “iTesla power systems library (iPSL): a modelica library for phasor time-domain simulations”. *SoftwareX* 2018;7:34–6. <https://doi.org/10.1016/j.softx.2018.01.002>. <https://www.sciencedirect.com/science/article/pii/S2352711018300050>.
- [12] Lara JD, Barrows C, Thom D, Krishnamurthy D, Callaway D. PowerSystems.jl — a power system data management package for large scale modeling. *SoftwareX* 2021;15:100747. <https://doi.org/10.1016/j.softx.2021.100747>. <https://www.sciencedirect.com/science/article/pii/S2352711021000765>.
- [13] de Castro M, Winkler D, Laera G, Vanfretti L, Dorado-Rojas SA, Rabuzin T, Mukherjee B, Navarro M. Version [OpenIPSL 2.0.0] - [iTesla power systems library (iPSL): a modelica library for phasor time-domain simulations]. *SoftwareX* 2023;21:101277. <https://doi.org/10.1016/j.softx.2022.101277>. <https://www.sciencedirect.com/science/article/pii/S2352711022001959>.
- [14] Lara JD, Henriquez-Auba R, Bossart M, Valverde G, Lincoln R, Roberts C, Krishnamurthy D, TagBot J, antonplietzsch. NREL-SIIP/PowerSimulationsDynamics.jl: v0.10.5. Zenodo. 2022. <https://doi.org/10.5281/zenodo.7072407>
- [15] Alsarayreh S, Dimitrovski R, Lekić A. Dynamic phasor state-space solver for hybrid AC/DC systems. In: Proceedings of the Power Systems Computation Conference (PSCC), Limassol, Cyprus; 2026. Accepted
- [16] Mirz M. A dynamic phasor real-time simulation based digital twin for power systems; 1. Auflage [Dissertation, Auch veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, RWTH Aachen University, 2020]. RWTH Aachen University, Aachen; 2020. <https://doi.org/10.18154/RWTH-2020-10412>. <https://publications.rwth-aachen.de/record/804608>.