Abstract—We describe the design, implementation, and experimental validation of an ultra-fast real-time hardware-in-the-loop emulation of a permanent magnet synchronous machine (PMSM) inverter drive. The power electronics converter and machine are modeled using a flexible piecewise linear state space approach and are simulated in hard real-time with \( 1 \mu \text{s} \) time step, which enables high-fidelity modeling of converter switching dynamics, including dead time and fully rectifying mode. We validate the fidelity of the real-time PMSM drive emulation by making real-time comparisons with a reference hardware model of a PMSM drive (scaled down system rated at 2.7 A and 200 V). Additionally, we experimentally demonstrate the capability of the hardware-in-the-loop PMSM drive emulation under various operating scenarios, including steady state, transient, and fault conditions.

I. INTRODUCTION

As the complexity of power electronics converters and control systems increases, there has been increasing demand for prototyping and validation tools that can reduce the development times of these systems while improving their reliability and fault tolerance. Recently, hardware-in-the-loop has demonstrated promise as a comprehensive tool for power electronics rapid prototyping and validation. Indeed, by interfacing the control system with a real-time emulation, hardware-in-the-loop enables engineers to quickly design and test controls without having to interface with a high-power power stage.

Hardware-in-the-loop for power electronics, particularly for PMSM drives, has been investigated in the past. Generally, these implementations fall into two categories: (1) PWM averaging modeling approach, and (2) hardcoded FPGA bitstream and I/O interface for a single topology.

For instance, in [1], the authors have investigated a PWM interrupt-driven design for a hardware-in-the-loop emulation of an automotive PMSM drive. They implement a gate signal averaging module that samples and averages the PWM \((f_s = 16 \text{ kHz})\) and calculates an estimation of the duty cycle. However, this implementation negates the ability to capture transients that occur on the time scale of the switching frequency (e.g. current and voltage ripple). Furthermore, these types of models are unable to capture effects such as dead time, fully rectifying mode, and some types of faults.

In [2], the authors have implemented an FPGA-based hardware-in-the-loop emulation of a PMSM motor drive system. The authors demonstrate the ability to prototype a closed-loop current and speed controller \((f_s = 5 \text{ kHz})\) for a PMSM drive, including the ability to model dead time and fully rectifying mode. However, in order to change circuit parameters or topology, the system requires a complete regeneration of the FPGA bitstream. This makes the process generally inflexible, as users will be unable to change and test circuit parameters without having to compile the entire FPGA firmware.

In this paper, we present a flexible, high-fidelity hardware-in-the-loop emulator for power electronics systems. The hardware-in-the-loop emulator leverages a novel FPGA-based soft processor that solves piecewise linear state space models in hard real-time with \( 1 \mu \text{s} \) time step [3]. The piecewise linear state space modeling approach enables complete modeling of all modes and transitions, including dead time, fully rectifying mode, and fault modes.

Additionally, the soft processor architecture of the FPGA enables users to load new circuit parameters or topologies without having to regenerate the FPGA bitstream. Indeed, the authors have developed a software toolchain that can automatically generate piecewise linear state space models, including transitions between modes, from a GUI-based circuit editor, which enables flexible and fast model generation.

This manuscript investigates the application of the hardware-in-the-loop emulator for a permanent magnet synchronous machine (PMSM) inverter drive. A high-performance PMSM drive requires closed-loop control of stator currents and rotor position or speed, as shown in Figure 1. In addition, fault tolerance and exhaustive testing of controls present significant challenges for many safety critical applications [4]. This paper develops and demonstrates a high-fidelity hardware-in-the-loop emulator that can accurately simulate the high frequency switching dynamics of a PMSM drive under steady state, transient, and fault conditions.

The paper is organized as follows. Section II develops the piecewise linear modeling techniques and the real-time implementation used for the PMSM drive emulation. Section III describes the experimental setup that is used to validate the fidelity of the real-time emulation. Section IV presents results that demonstrate the fidelity and capability of the real-time PMSM drive emulation as a hardware-in-the-loop prototyping platform. Section V concludes the paper.
II. MODELING APPROACH AND REAL-TIME IMPLEMENTATION

In this section, we describe the modeling approach, parameter identification techniques, and real-time processor platform used to implement the hardware-in-the-loop emulation of a power electronics system. While the modeling approach and real-time processor platform are general, we will, for the purposes of this paper, describe the implementation and validation techniques for a two-level, three-phase inverter drive connected to a PMSM.

First, we describe the piecewise linear state space modeling approach used to describe the two-level inverter and PMSM. Next, we detail the parameter identification techniques used to characterize a real hardware PMSM that is utilized as a reference model and for model fidelity validation. Lastly, we describe the hardware-in-the-loop implementation of the PMSM drive model on a custom-designed real-time computational platform.

A. PMSM drive modeling

We use a piecewise linear state space approach to model the power electronics converter and machine [5]. We model semiconductor devices (e.g. IGBTs, diodes, thyristors) as ideal switches, and generate a complete set of state space matrices that describes the dynamics of every mode, i.e. switch combination. Transitions between these modes are well-defined as a function of state variables or inputs to the model. The piecewise linear state space models, including transitions between modes, are automatically generated by a custom GUI-based software toolchain.

The topology of the PMSM drive is a two-level, three phase inverter, as shown in Figure 1. Each switch of the inverter is modeled as an IGBT with an anti-parallel diode. The modeling approach enables accurate emulation of the dead time, zero vector switching, fully rectifying mode, reverse diode recovery, and fault conditions, including shoot-through faults, open-phase faults, and individual switching device faults. It is worth noting that a simple voltage source model of an inverter would be unable to capture many of these operating and fault conditions.

Then, we use a well-known state space model to describe the PMSM using two current vectors in the $d$-$q$ axis [6]. In the $d$-$q$ rotor reference frame, the model for the PMSM is as follows:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & 0 \\ 0 & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{1}{L_d} \begin{bmatrix} 0 & L_d \\ L_q & 0 \end{bmatrix} \begin{bmatrix} v_d \\ E_q \end{bmatrix}$$

$$m_e = \frac{3}{2} p [\lambda_{pm} i_q + (L_d - L_q) i_d i_q]$$

where $E_d = \lambda_{pm} \cdot \omega \sin(\theta)$, $E_q = \lambda_{pm} \cdot \omega \cos(\theta)$, and $m_e$ is the electromagnetic torque.

In Section IV, we will present results that demonstrate the fidelity of this model under steady state, transient, and fault conditions.

B. PMSM parameter identification

We use the Anaheim Automation EMJ-04APB PMSM (rated at 2.7 A and 200 V) as a scaled down reference machine to verify the fidelity of the model developed in Section II-A. Details of this machine can be found in Table I. In order to model this machine according to the method presented in Section II-A, there are three parameters from the PMSM that require identification: the stator resistance $R_s$, the magnet flux linkage $\lambda_{pm}$, and the $d$- and $q$-axis inductances $L_d$ and $L_q$. These parameters are identified off-line prior to real-time emulation. The steps we use to measure these parameters follow closely to the methodology proposed in [7].

First, to identify the stator resistance $R_s$, we sweep the DC current on each line-to-line terminal from 0 to 5 A while measuring the resulting voltage. Using this method, we determined that the average stator resistance per phase is $R_s = 2.174 \Omega$.

Second, we identify the magnet flux linkage $\lambda_{pm}$ by measuring the open-circuit back EMF voltage. Since the line-to-neutral back EMF voltage $V_s$ is linearly proportional to the electrical frequency of the PMSM $\omega$ by the relationship:

<table>
<thead>
<tr>
<th>PMSM ratings</th>
<th>Anaheim EMJ-04APB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model identifier</td>
<td>Anaheim EMJ-04APB</td>
</tr>
<tr>
<td>Stator resistance, $R_s$</td>
<td>2.174 Ω (from Sec. II-B)</td>
</tr>
<tr>
<td>Magnet flux linkage, $\lambda_{pm}$</td>
<td>0.3859 Wb (from Sec. II-B)</td>
</tr>
<tr>
<td>$d$-axis inductance, $L_d$</td>
<td>2.5 mH (from Sec. II-B)</td>
</tr>
<tr>
<td>$q$-axis inductance, $L_q$</td>
<td>2.5 mH (from Sec. II-B)</td>
</tr>
<tr>
<td>Number of pole pairs</td>
<td>2</td>
</tr>
<tr>
<td>Base speed</td>
<td>5000 r/min</td>
</tr>
<tr>
<td>Phase current</td>
<td>2.7 A</td>
</tr>
<tr>
<td>Phase voltage</td>
<td>200 V</td>
</tr>
<tr>
<td>Real-time computational platform</td>
<td>Xilinx Virtex-5 ML506</td>
</tr>
<tr>
<td>FPGA device</td>
<td>Xilinx Virtex-5 ML506</td>
</tr>
<tr>
<td>Clock speed</td>
<td>100 MHz</td>
</tr>
<tr>
<td>ADC sampling rate</td>
<td>1 MSPS</td>
</tr>
<tr>
<td>DAC sampling rate</td>
<td>1 MSPS</td>
</tr>
<tr>
<td>Inverter module ratings</td>
<td>20 A, 600 V</td>
</tr>
<tr>
<td>SVM PWM generator</td>
<td></td>
</tr>
<tr>
<td>Switching frequency</td>
<td>4 kHz</td>
</tr>
<tr>
<td>Dead time</td>
<td>1 μs</td>
</tr>
</tbody>
</table>

TABLE I

SPECIFICATIONS AND RATINGS FOR EXPERIMENTAL TESTBED

We use a piecewise linear state space approach to model the power electronics converter and machine [5]. We model semiconductor devices (e.g. IGBTs, diodes, thyristors) as ideal switches, and generate a complete set of state space matrices that describes the dynamics of every mode, i.e. switch combination. Transitions between these modes are well-defined as a function of state variables or inputs to the model. The piecewise linear state space models, including transitions between modes, are automatically generated by a custom GUI-based software toolchain.

The topology of the PMSM drive is a two-level, three phase inverter, as shown in Figure 1. Each switch of the inverter is modeled as an IGBT with an anti-parallel diode. The modeling approach enables accurate emulation of the dead time, zero vector switching, fully rectifying mode, reverse diode recovery, and fault conditions, including shoot-through faults, open-phase faults, and individual switching device faults. It is worth noting that a simple voltage source model of an inverter would be unable to capture many of these operating and fault conditions.

Then, we use a well-known state space model to describe the PMSM using two current vectors in the $d$-$q$ axis [6]. In the $d$-$q$ rotor reference frame, the model for the PMSM is as follows:

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} -\frac{R_s}{L_d} & 0 \\ 0 & -\frac{R_s}{L_q} \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \frac{1}{L_d} \begin{bmatrix} 0 & L_d \\ L_q & 0 \end{bmatrix} \begin{bmatrix} v_d \\ E_q \end{bmatrix}$$

$$m_e = \frac{3}{2} p [\lambda_{pm} i_q + (L_d - L_q) i_d i_q]$$

where $E_d = \lambda_{pm} \cdot \omega \sin(\theta)$, $E_q = \lambda_{pm} \cdot \omega \cos(\theta)$, and $m_e$ is the electromagnetic torque.

In Section IV, we will present results that demonstrate the fidelity of this model under steady state, transient, and fault conditions.

B. PMSM parameter identification

We use the Anaheim Automation EMJ-04APB PMSM (rated at 2.7 A and 200 V) as a scaled down reference machine to verify the fidelity of the model developed in Section II-A. Details of this machine can be found in Table I. In order to model this machine according to the method presented in Section II-A, there are three parameters from the PMSM that require identification: the stator resistance $R_s$, the magnet flux linkage $\lambda_{pm}$, and the $d$- and $q$-axis inductances $L_d$ and $L_q$. These parameters are identified off-line prior to real-time emulation. The steps we use to measure these parameters follow closely to the methodology proposed in [7].

First, to identify the stator resistance $R_s$, we sweep the DC current on each line-to-line terminal from 0 to 5 A while measuring the resulting voltage. Using this method, we determined that the average stator resistance per phase is $R_s = 2.174 \Omega$.

Second, we identify the magnet flux linkage $\lambda_{pm}$ by measuring the open-circuit back EMF voltage. Since the line-to-neutral back EMF voltage $V_s$ is linearly proportional to the electrical frequency of the PMSM $\omega$ by the relationship:

\[
V_s = \frac{\lambda_{pm}}{\pi f}
\]

where $\lambda_{pm}$ is the magnet flux linkage, $f$ is the electrical frequency, and $\pi$ is the mathematical constant. By measuring the voltage $V_s$ at different frequencies $f$, we can solve for $\lambda_{pm}$.

Finally, we identify the $d$- and $q$-axis inductances $L_d$ and $L_q$ using standard electrical measurements. The inductances can be determined from the current $i_d$ and $i_q$, and voltage $v_d$ and $v_q$ as follows:

\[
L_d = \frac{v_d}{i_d}
\]

\[
L_q = \frac{v_q}{i_q}
\]
V_s = \lambda_{pm} \cdot \omega$, we determined that the magnet flux linkage is \( \lambda_{pm} = 0.3859 \) Wb, as shown in Figure 2a.

Third, we experimentally identify the \( d \)- and \( q \)-axis inductances. We used a locked-rotor test in which a sinusoidal voltage excitation is applied to a single phase of the PMSM. In this configuration, we can calculate the \( d \)- and \( q \)-axis flux linkages \( \lambda_{d,q} \) due to the resulting sinusoidal current since 
\[
\lambda_{d,q} = \frac{1}{2} \int_0^T (V - \frac{3}{2} R_s I_{d,q}) \, dt
\]
and 
\[
L_{d,q} = \lambda_{d,q} \cdot I_{d,q}^{-1}
\]
Also, since the reference PMSM has a cylindrical rotor, it follows that 
\( L_d = L_q \). The \( d \)- and \( q \)-axis inductances as a function of a function of peak applied current is shown in Figure 2b.

Because our implementation of the PMSM model in Section II-A requires constant \( L_d \) and \( L_q \), we determined a function of the form 
\[
L_{d,q} = k_1 \tan^{-1}(k_2 I_{d,q}) + k_3
\]
that models the saturation curve seen in Figure 2b. Using this function, we select a peak current operating point to test, and we simulate a model of the PMSM with the appropriate value of \( L_d \) and \( L_q \).

C. Hardware-in-the-loop implementation

The piecewise linear state space model of the two-level inverter and PMSM is implemented on a real-time processor platform [5]. The processor platform is general such that it can execute in real-time any power electronics system described as a piecewise linear state space model. The soft processor architecture enables users to easily load new circuit parameters or topologies without having to regenerate the FPGA bitstream.

The processor is based on the Xilinx Virtex-5 ML506 FPGA, and calculates the entire power stage model in hard real-time with 1 \( \mu \)s time step, including input-output latency. Additionally, the processor handles transitions between each mode depending on inputs to the system (e.g. gate drive signals, contactor signals) or internal state variables (e.g. inductor currents). During real-time execution, the appropriate submodel state space mode is calculated by a highly optimized linear solver.

We use a separate DSP-based Texas Instruments TMS320F2808 Control Card to implement a closed-loop control algorithm for the PMSM real-time model. The controller enables closed-loop control of rotor speed and \( d \)- and \( q \)-axis currents by monitoring the phase currents and rotor position and speed from the real-time model.

III. HARDWARE REFERENCE MODEL SETUP FOR PMSM INVERTER DRIVE

In order to comprehensively validate the fidelity of the real-time emulation, we created a validation setup that runs the real-time emulation in parallel with an identical power stage implementation of a PMSM drive. The validation setup is shown in Figure 3. Ratings and details of the hardware reference model can be found in Table I.

The hardware reference model consists of custom-designed back-to-back two-level, three-phase inverter drive rated at 20 A and 600 V. For this paper, only one of the two inverters was used. An external DC power supply was used to supply the DC bus. The inverter drive integrates measurements for phase currents, phase and bus voltages, and temperature. Contactors are located on all three output phases and provide open-phase fault injection capability. Additionally, an on-board CPLD enables programmable signal routing, fault injection, and fault protection.

As mentioned previously, the reference hardware PMSM used for validation is the Anaheim Automation EMJ-04APB. In the validation setup, the PMSM is coupled to a load machine controlled by an ABB ACS800-U11 direct torque control (DTC) drive, which enables precise control of the mechanical torque load on the PMSM.

In order to make real-time comparisons, gate drive signals from the TMS320F2808 controller are routed in parallel to the real-time simulator and the hardware reference model. Phase current and rotor position information from the hardware reference model are used for closed-loop control, as shown in Figure 3.
IV. EXPERIMENTAL VERIFICATION AND RESULTS

Using the testbed described in Section III and shown in Figure 3, we validate the fidelity of the real-time PMSM drive emulation by making real-time comparisons with the hardware reference model under both steady state and transient operating conditions. Additionally, we demonstrate the hardware-in-the-loop capability of the real-time emulation under fault conditions by injecting open-phase faults into the PMSM drive.

A. Model validation

First, we will validate the fidelity of the real-time emulation by making steady state comparisons with the hardware reference model. We define the rotor speed reference \( \omega_{\text{ref}} \) and the \( d \)-axis current reference via the closed-loop controller on the TMS320F2808 Control Card. The reference set points are defined as \( \omega_{\text{ref}} = 0.2 \text{ pu} \) and \( i_{\text{ref}} = 0.2 \text{ pu} \). The drive is allowed to reach steady state with no load, and we compare line-to-line voltage \( v_{ab}(t) \), as shown in Figure 4, and the phase current \( i_a(t) \), as shown in Figure 5.

As seen in both steady state figures, the real-time emulation of the PMSM drive closely matches the response of the hardware reference model. In particular, Figure 4b demonstrates the capability of the real-time emulation to closely match the voltage waveform of the hardware reference model on the scale of the switching frequency. This highlights both the speed and low latency of the hardware-in-the-loop platform. Additionally, in Figure 5b, we can observe the small phase current error, approximately 3.90%, between the RMS of the real-time emulation and the hardware reference model. Reasons for this discrepancy are discussed below.

Next, we will validate the fidelity of the real-time emulation during a transient case in which an external mechanical torque load step is applied to the shaft of the PMSM. We use a load machine controlled by an ABB ACS800-U11 direct torque control (DTC) drive to apply a 0 to 2 Nm mechanical torque step. A synchronized and identical torque step is applied to the
real-time emulation. The torque step is applied when $\omega_{\text{ref}} = 0.3 \text{ pu}$ and the PMSM is running at steady state.

Figure 6 shows the transient response of the phase current to application of the torque step. As seen in Figure 6a, the real-time emulation closely matches the response of the hardware reference model in both steady state and transient conditions, which validates the capability of the hardware-in-the-loop platform to accurately emulate these operating conditions for a PMSM drive. In Figure 6b, we can observe the small phase current error of 3.90% before the application of the torque step, and a slight increase in phase current error to 4.92% after the application of the torque step.

The authors attribute the real-time emulation discrepancies to the modeling implementation of the PMSM. The implementation described in Section II-A does not have the capability to account for parameters that vary as a function of current, flux, and/or temperature. In future works, the authors will investigate PMSM emulation methods that account for varying machine parameters, such as $d$- and $q$-axis inductions.

### B. Hardware-in-the-loop open-phase fault

Lastly, we investigated the open-phase fault condition in the real-time emulation of the PMSM drive. We introduced contactors in the real-time model to dynamically inject open-phase faults in each of the three phases. Table II shows the expected $\alpha$-$\beta$ current response during an open stator condition.

<table>
<thead>
<tr>
<th>Phase</th>
<th>Expected Current Response</th>
</tr>
</thead>
</table>
| $a$ open stator: | $i_\alpha = 0$
| | $i_\beta = \sqrt{2}i_b$
| $b$ open stator: | $i_\alpha = \sqrt{2}i_a$
| | $i_\beta = \sqrt{2}i_a$
| $c$ open stator: | $i_\alpha = \sqrt{2}i_a$
| | $i_\beta = -\sqrt{2}i_a$

Table II: Expected $\alpha$-$\beta$ Current Response for PMSM Drive Open-Phase Fault

Fig. 7. Real-time emulation of PMSM $\alpha$-$\beta$ current loci in normal and fault conditions.

Additionally, we demonstrated the capability of the real-time PMSM emulation as a hardware-in-the-loop prototyping platform by validating the open-phase fault condition.

The key advantages of the proposed hardware-in-the-loop platform are twofold. First, the piecewise linear state space modeling approach enables comprehensive modeling of power electronics, including dynamics that occur on the scale of the switching frequency. Second, the flexible modeling implementation enables the platform to be used as a powerful tool for power electronics rapid prototyping and validation.

V. CONCLUSIONS

In this manuscript, we presented modeling and implementation techniques for a real-time PMSM inverter drive emulation. We validated the fidelity of the real-time emulation by making steady state and transient comparisons with a hardware reference model. We provided detailed $\mu$s-scale waveform comparisons that demonstrated the fidelity of the real-time emulator at the time scale of the converter switching frequency.

### REFERENCES


