

Three-Phase PWM Schemes and Impacts on Motor Condition Monitoring

Anish Sivakumar, Dr. Ilamparithi T

University of Victoria, Department of Electrical and Computer Engineering

March 20, 2024

BACKGROUND

Electric motors can develop various faults over time, affecting performance and reliability. These include rotor eccentricity, magnet faults, open and short-circuit stator windings, and over temperature [1].

- Rotor eccentricity occurs when the rotor deviates from the stator's centerline, leading to uneven gaps and mechanical stress.
- Magnet faults weaken the magnetic field, reducing torque and efficiency.
- Stator winding faults disrupt current flow, causing overheating and in severe cases, complete motor failure.

Monitoring and detecting motor faults is crucial to prevent costly downtime and equipment damage. Techniques like vibration analysis, thermal sensing, and electrical measurements are commonly used to monitor motor condition [2].

- Current signature analysis offers a non-intrusive approach to detect motor abnormalities [2].

Three-phase pulse-width-modulation (PWM) is used to drive industrial motors by converting DC to three-phase AC through an inverter circuit. Different PWM techniques impact factors such as harmonic distortion, torque ripple, efficiency, and motor performance across various operating conditions.

OBJECTIVE

Most current monitoring methods are studied and developed using Sinusoidal PWM to control the current and voltage fed to an inverter-driven three-phase motor.

This project aims to investigate the differences between popular three-phase PWM schemes to understand how they may impact motor condition monitoring techniques that rely on inverter and motor current data. The PWM methods of interest are listed below.

1. Sinusoidal PWM
2. Third-Harmonic Injection PWM
3. Space Vector PWM

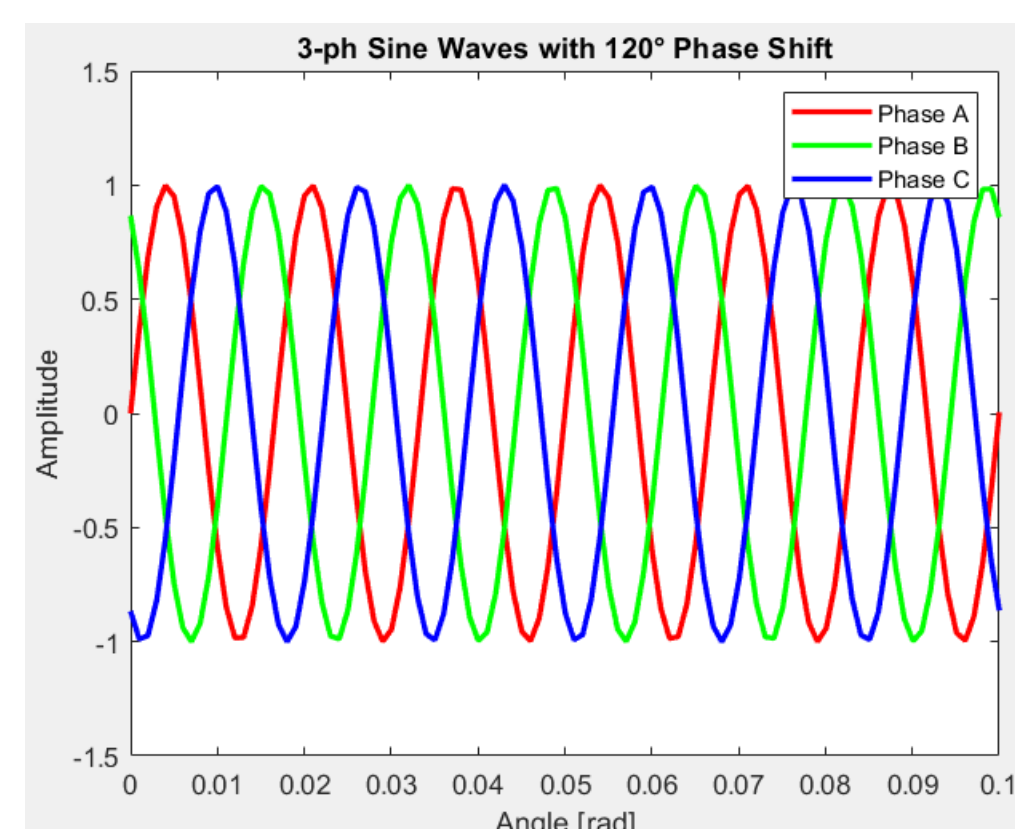


Figure 1: Three-phase sine waveforms

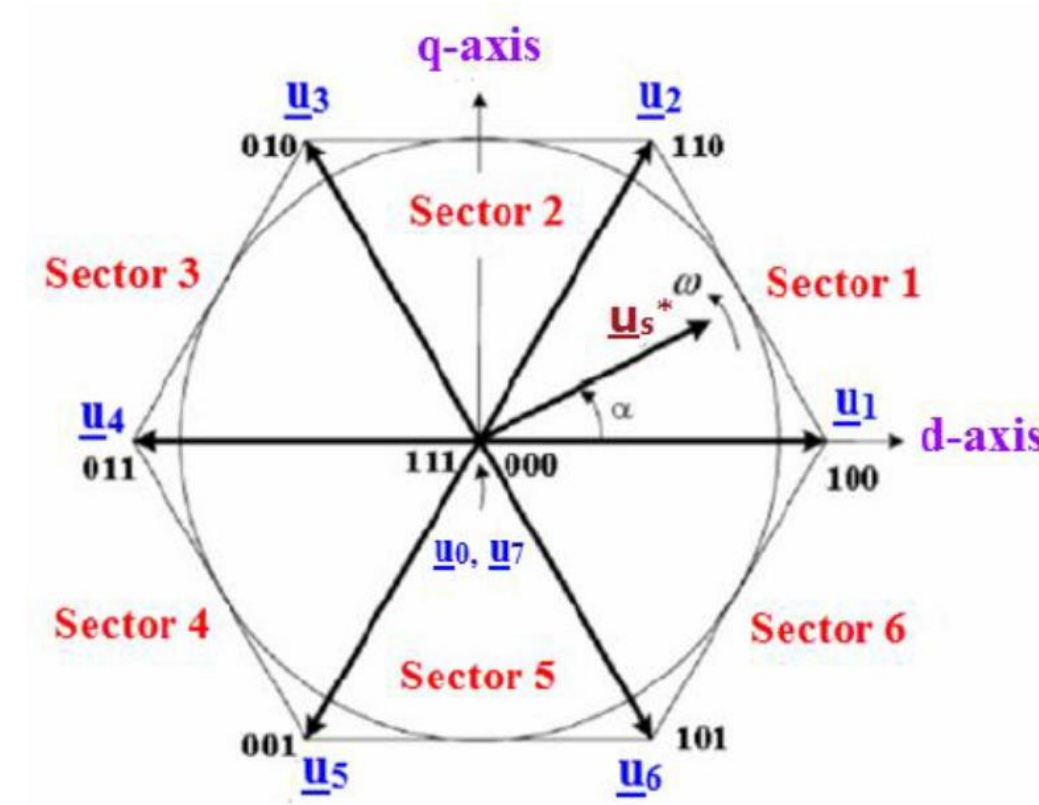


Figure 2: Space Vector PWM diagram [3]

PROPOSED SOLUTION

The proposed solution is to simulate the PWM techniques in Simulink and implement them on a low-cost microcontroller.

- Simulink models using the Simscape Electrical library were created to represent a physical test system and the modulation technique.
- Sinusoidal PWM was implemented on the Teensy 3.2 microcontroller board because of its Arduino IDE compatibility and ease of set-up. The onboard ARM Cortex-M4 MCU has a clock speed of 72MHz and 34 digital I/O pins [4].

SIMULATION

50Hz sinusoidal PWM with 1kHz switching frequency was simulated.

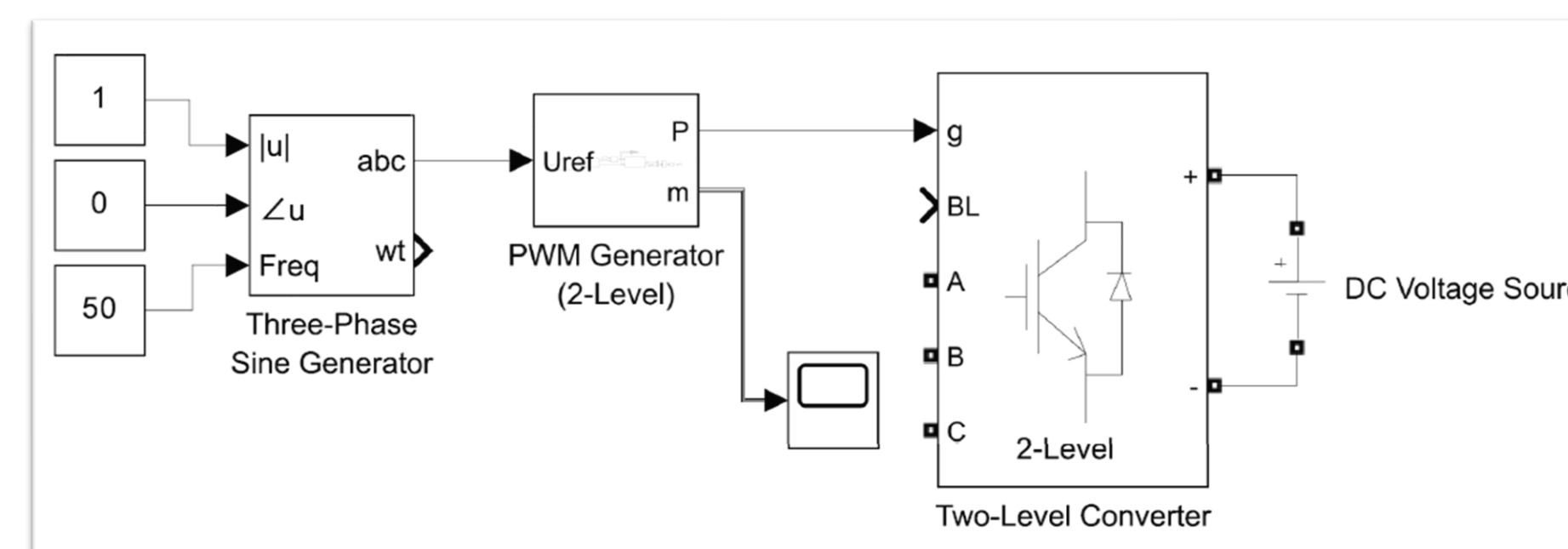


Figure 3: Sinusoidal PWM Simulink model

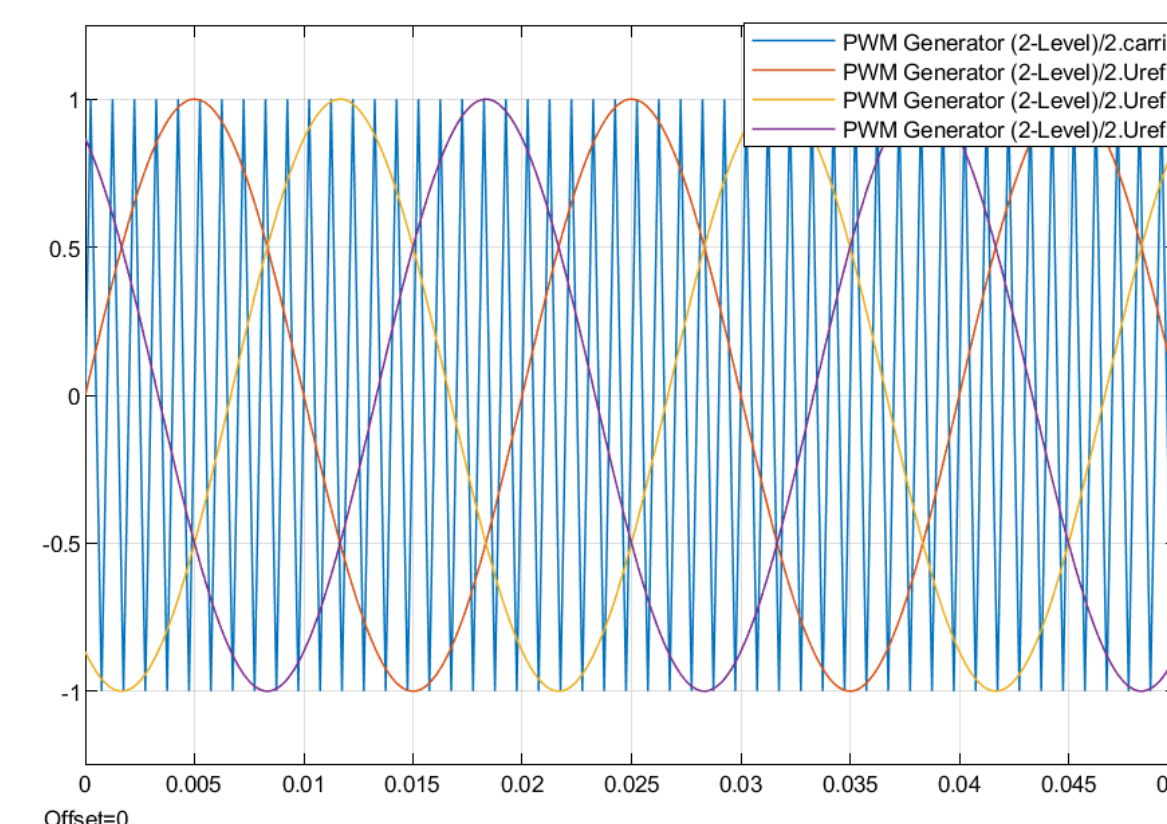


Figure 4: Sinusoidal PWM waveforms

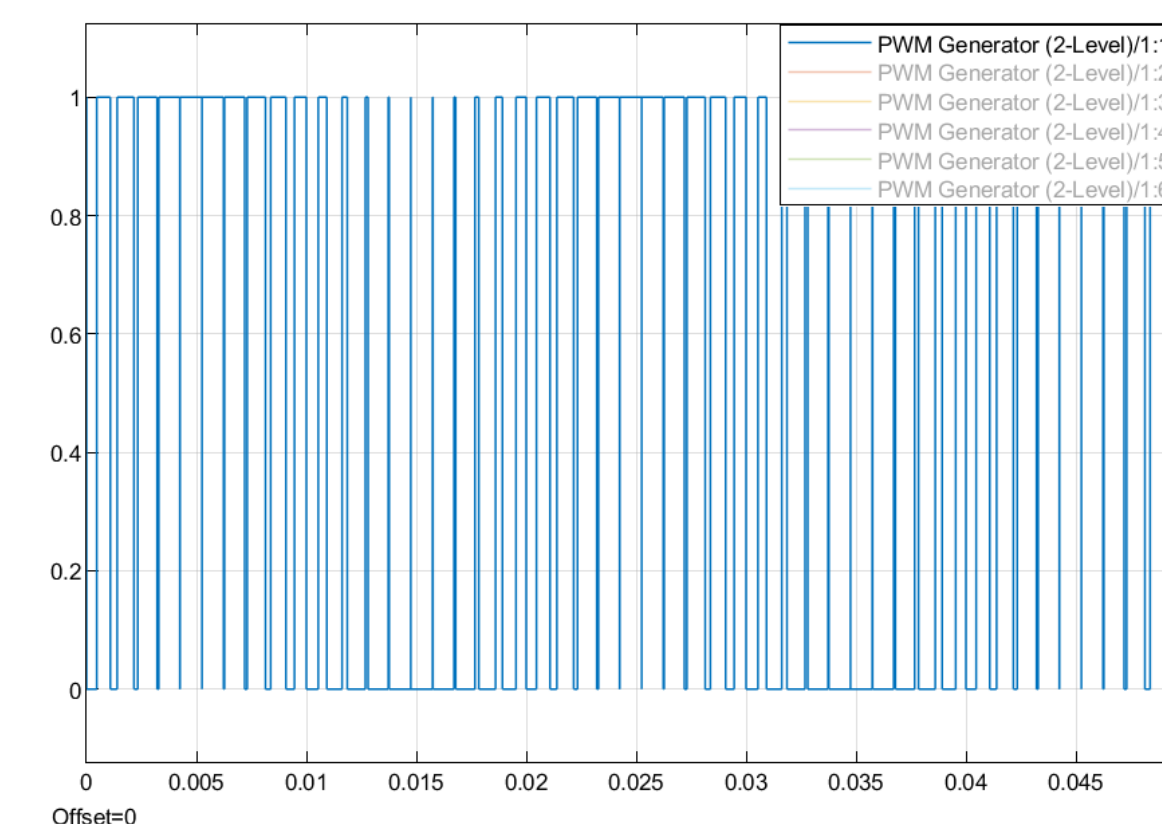


Figure 5: Sinusoidal PWM switching output

50Hz sinusoidal PWM with 1kHz switching frequency was simulated with the third-harmonic injection.

- Modulation index (m) can be increased to 1.154 with third harmonic addition amplitude of $\frac{1}{6} \times m$ [5]
- Third-harmonic injection increases DC bus utilization with "double humps" in sine wave

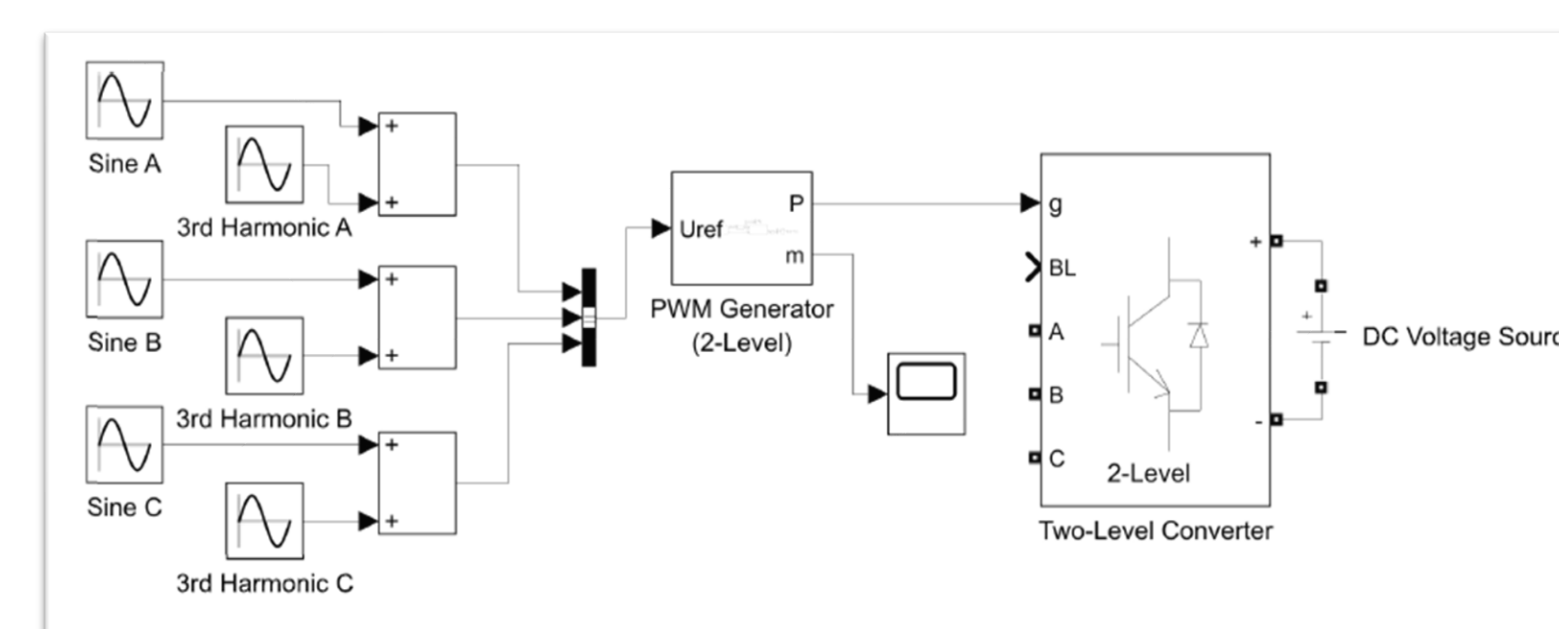


Figure 6: Third-Harmonic PWM Model

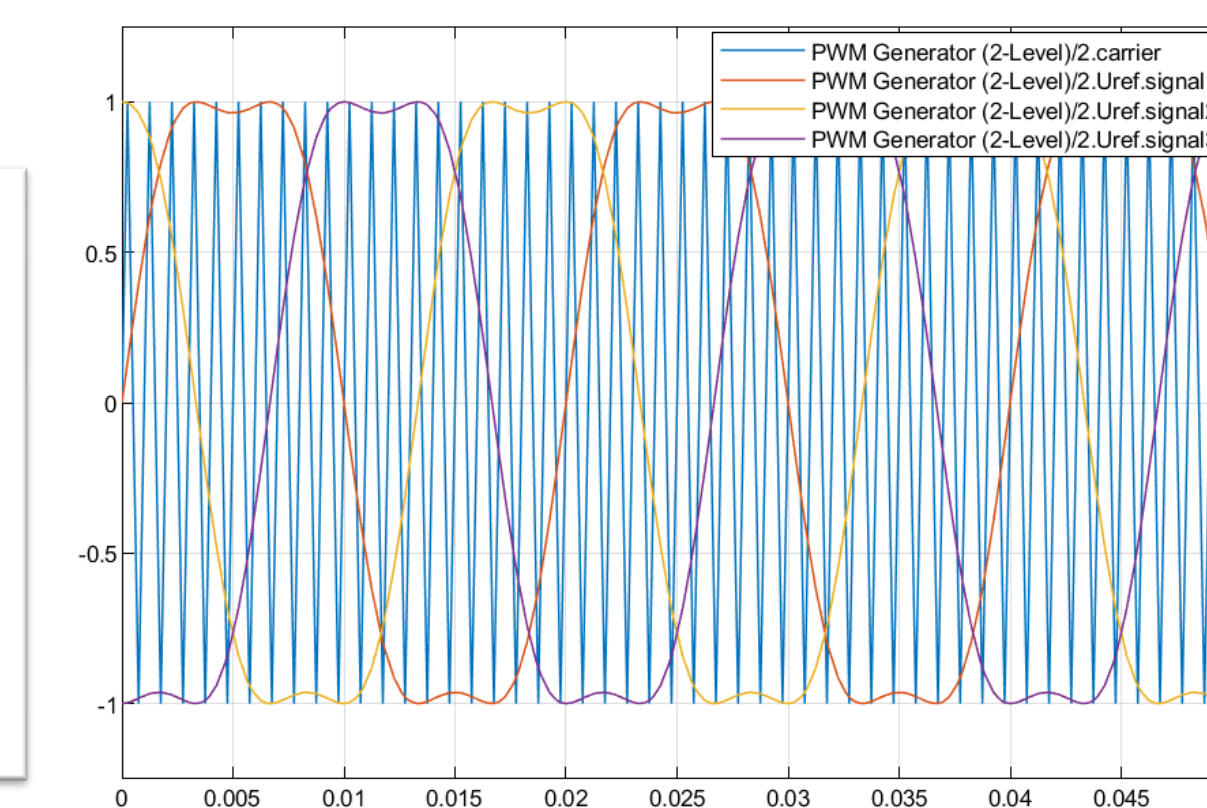


Figure 7: Third-Harmonic PWM waveforms

IMPLEMENTATION

To replicate sinusoidal PWM with a hardware-limited microcontroller, a simple interrupt service routine was written in Arduino IDE to update PWM values on six digital output pins according to a predefined table.

- Fixed-frequency 50Hz sine function and 1kHz triangular carrier function were discretized and compared in Python to generate a 1000-value binary PWM table. The binary values corresponded to one switching device being turned on or off.
- 3-phase sinusoidal PWM was implemented with a 50kHz interrupt service routine that updates six digital output pin values using the PWM look-up table. Output values for all three phases, roughly 120° phase shifted, were determined by indexing the table at three points $\frac{1}{3}$ of the table size shifted from each other.

RESULTS

- The rise and fall times from the microcontroller digital output signals were measured with an oscilloscope to be fluctuating around $1\mu s - 3\mu s$.
- This time is too large to maintain a safe transistor dead-time with satisfactory motor performance. Interestingly, this behavior varied when outputting a different set of table values.
- A deadtime within nanoseconds is desired to replicate standard motor usage and consequently, the software was not tested with an inverter and motor in the laboratory.

NEXT STEPS

- Next steps are to implement the PWM methods on a more robust microcontroller such as a STM32 board to be able to experiment on an inverter and motor.
- Current waveforms from the different PWM techniques can be compared against each other and as inputs into a motor condition monitoring model.

ACKNOWLEDGEMENTS

- This research was supported by the Jamie Cassels Undergraduate Research Awards, University of Victoria, and supervised by Dr. Ilamparithi T (Department of Electrical and Computer Engineering).