

ADAPTIVE RANDOM PULSEWIDTH MODULATION SCHEME FOR LOCATING AND SHAPING THE
SPECTRAL CONTENT OF INDUCED VIBRATION IN ELECTRIC POWER CONVERSION, APPLIED TO A
THREE-PHASE SYNCHRONOUS MOTOR

By

Jonathan H. Johnston

A THESIS

Submitted to
Michigan State University
In partial fulfillment of the requirements
For the degree of

MASTER OF SCIENCE

Electrical Engineering

2012

ABSTRACT

ADAPTIVE RANDOM PULSEWIDTH MODULATION SCHEME FOR LOCATING AND SHAPING THE SPECTRAL CONTENT OF INDUCED VIBRATION IN ELECTRIC POWER CONVERSION, APPLIED TO A THREE-PHASE SYNCHRONOUS MOTOR

By

Jonathan H. Johnston

Pulsewidth modulation schemes used in various electric power conversion applications can result in objectionable vibration and acoustic behavior resulting from electromagnetically induced mechanical response. The response will be more salient if the modulation carrier frequency coincides with a system mechanical resonance, especially if the carrier frequency is constant. Several random pulsewidth modulation (RPWM) techniques have been proposed over the last 20 years in order to reduce the prominence of the modulation over the 2 kHz to 5 kHz range. The area where these RPWM techniques are lacking arises when system response varies under different operating conditions, or when it is not known a priori. While effective at reducing objectionable content (either acoustic or electromagnetic) by spreading it over neighboring regions, they still do not address the robustness requirements of systems with varying responses. Additionally, the approach may actually increase the likelihood of coinciding with a system resonance, since the exciting frequency range is wider than in constant switching frequency techniques. The proposed approach builds upon existing RPWM strategies by using an adaptive algorithm to position and shape the RPWM modulation frequency distribution based on system vibration feedback. Gradient descent is employed to vary the center frequency and distribution range of the randomized carrier frequency in response to change in the objective metric based on the vibration feedback.

Dedication

To Kristina, Carter, and Audrey

Table of Contents

List of Figures	vi
Key to Abbreviations.....	xii
Introduction	1
Literature Survey.....	2
Non-RPWM Modulation Techniques	3
RPWM Techniques	5
Previous Adaptive Approaches Relevant to this Topic	8
Proposed Technique	9
Gradient Descent.....	11
Convergence and Stability and Variable Learning Rate	14
Cost Considerations.....	27
Implementation.....	28
Software Overview	28
Design of Numerical Representation and Loop Rates.....	29
Hardware overview	39
Modeling and Simulation	40
Inverter Model	40
Deadtime Model.....	42
Motor Model	45
Simulation Comparison of SPWM and various RPWM techniques	48
Simulation of SPWM and RPWM behavior with Deadtime	52
Simulation of ARPWM.....	61
Experiments and Results.....	65
Conclusions	83
APPENDICES	85

Appendix A: Arbitrary Motor Control Algorithm Prototyping Hardware	86
Appendix B: Software Details	90
FPGA Code	90
Host Code	99
BIBLIOGRAPHY	104

List of Figures

Figure 1. SPWM with a sinusoidally varying carrier frequency. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis.....	3
Figure 2. SPWM with carrier function whose frequency is a function of the slope of the reference signal. Although typically employed in trapezoidal/6-step controllers, here the concept is portrayed in an SPWM context.	4
Figure 3. MI, Carrier, Switching Command S (left column), and PSD of S (right column) for SPWM (top), RPWM with random carrier frequency (2nd from top), RPWM with random carrier function (3rd from top), RPWM with random pulse position (bottom).....	6
Figure 4. Typical search space with constant range and center frequency cross sections indicated with lines. Signatures from these cross sections will be used for our analysis.	15
Figure 5. A constant center frequency slice signature from the search space	16
Figure 6. A constant range slice signature from the search space	16
Figure 7. Block diagram of ARPWM with gradient descent.....	18
Figure 8. Response and Range versus learning rate versus iteration number (starting at a high-response location) for the response signature as range is varied over a constant distribution center frequency.....	20
Figure 9. Response and Range versus learning rate versus iteration number (starting at a high-response location) for the response signature as center frequency is varied over a constant distribution range (starting somewhere in the local minimum at 3400Hz).	21
Figure 10. Analysis of local minima for typical response vs range for a constant center frequency slice	24
Figure 11. Analysis of local minima for typical response vs center frequency for a constant range slice	25
Figure 12. Adaptive RPWM (ARPWM) FPGA implementation	28

Figure 13. PWM switch command S , carrier, and reference signals versus control loop iteration with discrete implementation. The actual transition times are dependent on our numeric representations and timing resolutions. 30

Figure 14. Experimental frequency resolution (or maximum error) versus switching frequency at 250ns control loop rate (left) and frequency resolution as a percentage of the switching frequency versus switching frequency (right). (when not constrained by insufficient numerical resolution of the carrier and reference signals) 31

Figure 15. Duty cycle error (%) versus switching frequency when $MI = 0.9$ 33

Figure 16. Peak duty cycle error (ppm) versus MI . At each MI , the frequency was varied from 1 kHz to 10 kHz and the maximum and minimum values were plotted. In this case the carrier frequency had no phase shift, so this is not representative of the worst case duty cycle error. 33

Figure 17. Peak duty cycle error versus carrier phase (evaluated with at 100 equally divided intervals). This just gives a rough idea of the error, since we skipped over several phase angles and several MI values in order to generate this graphic in a timely manner..... 34

Figure 18. 64-bit, 250ns (left) , 16-bit, 250ns (right). The resolution of the 16-bit carrier frequency is insufficient; the frequency resolution is constant from 1000 to 10000Hz since the carrier resolution is determining the overall frequency resolution. The 64-bit representation offers no improvement over the 32-bit representation since both resolutions are more than sufficient, making frequency resolution completely dependent on control loop rate. 36

Figure 19. The 16-bit representation is does not provide sufficient resolution, while the 64-bit representation is not required since 32 bits provide all of the necessary resolution needed for our 4 MHz control loop rate. 36

Figure 20. 25ns control loop period (40 MHz), 32-bit (left) and 64-bit (right). Even at 40 MHz, the loop rate seems to be the limiting factor on the frequency resolution (based on the fact that the shape of the curve conforms to the equation relating frequency resolution to control loop rate when not constrained by numerical representation resolution)..... 37

Figure 21. To better illustrate the effect of numerical representation and loop rate on the frequency resolution, the frequency resolution as a percentage of switching frequency is plotted versus switching frequency. 25ns control loop period, 32-bit (left) and 64-bit (right).. 38

Figure 22. Picture of the entire system	39
Figure 23. Topology of 3-phase inverter used in this study.	41
Figure 24. Inverter Model, Ideal Switches and Diodes	41
Figure 25. Illustration of inverter behavior during transitions in which deadtime influences the actual output voltage and introduces distortion. Each step (labeled A through J) is mapped to its corresponding state, indicating which component (if any) is conducting.	43
Figure 26. 3-phase synchronous motor model equivalent circuit.....	46
Figure 27. Example simulation time histories.....	47
Figure 28. Frequency-domain phase current (left) and time-domain phase current (right): SPWM @ 4 kHz	48
Figure 29. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random carrier frequency, 4 kHz center frequency, 1 kHz range.	49
Figure 30. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random carrier function, 4 kHz center frequency, 10 kHz instantaneous range. The range is larger for the random function because the average range over a switching cycle is much less than the instantaneous range.....	50
Figure 31. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random pulse position, 4 kHz center frequency, random selection of leading or trailing on-time transition.....	51
Figure 32. Frequency-domain phase current (left) and time-domain phase current (right): SPWM @ 4 kHz, 120 rad/s fundamental frequency, 500ns Deadtime	53
Figure 33. Notice the extra content at multiples (5 th order at about 100 Hz and 7 th order at about 140 Hz) of the fundamental (~20 Hz) due to deadtime distortion with SPWM.	54
Figure 34. Frequency-domain phase current (left) and time-domain phase current (right): SPWM @ 4 kHz, 60 rad/s fundamental frequency, 1000ns Deadtime	55

Figure 35. Frequency-domain phase current (left) and time-domain phase current (right): SPWM @ 4 kHz, 60 rad/s fundamental frequency, 4000ns Deadtime	56
Figure 36. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random carrier frequency, 4 kHz center frequency, 1 kHz range, 60 rad/s fundamental frequency, 4000ns Deadtime	57
Figure 37. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random pulse position, 4 kHz center frequency, 60 rad/s fundamental frequency, 4000ns Deadtime. Notice the improvement in distortion.	58
Figure 38. The Difference in phase current amplitude between simulations performed at various deadtime values and zero deadtime. RPWM with random carrier frequency, 4 kHz center frequency with 1 kHz range.	59
Figure 39. Example simulated system response (left) and corresponding search space (right). The search space is the value of our objective function at each center frequency/range combination over the frequency range of interest.	62
Figure 40. Simulated ARPWM final steady state distribution in presense of smooth valley with moderate response levels in search space.	63
Figure 41. Simulated ARPWM final steady state distribution in presense of sharp valley with low response levels in search space.	64
Figure 42. Simulated ARPWM final steady state distribution after sharp valley is adjusted to flat valley and center of valley is shifted above 3kHz.	64
Figure 43. The program written to iterate through and map the search space.	65
Figure 44. Search space at 10 Hz fundamental frequency, no load	66
Figure 45. ARPWM over time. 11 Hz fundamental frequency.	67
Figure 46. Search space, 20 Hz fundamental, no load.....	68
Figure 47. ARPWM over time at 20 Hz fundamental.....	69

Figure 48. Map of search space generated when the mapping procedure was repeated while allowing the range to increase up to 3 kHz.	70
Figure 49. Close-up of system response near a resonant peak (left) and frequency slice response versus range (right).	70
Figure 50. Best-case initial carrier frequency distribution conditions. Since the initial conditions were near-optimal in a smooth region of the search space, there wasn't much searching to do.	72
Figure 51. Moderate-case initial carrier frequency distribution conditions. Notice how the range decreases over the first 13 seconds; the search space is not so trivial in this region and the gradient descent algorithm is misled for a while.	73
Figure 52. Worst-case initial carrier frequency distribution conditions.	74
Figure 53. Temporal behavior of ARPWM with a constant learning rate of 0.01.	76
Figure 54. Temporal behavior of ARPWM with a constant learning rate of 0.02.	77
Figure 55. Temporal behavior of ARPWM with the variable learning rate algorithm	78
Figure 56. Variable learning rate metrics over time during execution of the variable learning rate algorithm.	79
Figure 57. Variable learning rate metrics over time during another execution of the variable learning rate algorithm.	81
Figure 58. The measurement and control system designed to provide a control algorithm prototyping system.	86
Figure 59. The voltage signal conditioning circuit.	87
Figure 60. Optocoupler circuit for digital signal isolation.	88
Figure 61. Reference design motor drive.	89
Figure 62. LabVIEW tm FPGA code Front Panel	90

Figure 63. FPGA Acquisition, Filtering, Downsampling, FFT, DMA and Storage.....	91
Figure 64. FPGA One-Sided Spectrum Transfer to Host	94
Figure 65. FPGA 'Next Carrier Frequency' Generation Thread.....	95
Figure 66. FPGA main RPWM control thread.	97
Figure 67. Main user interface.....	99
Figure 68. The alternating gradient descent algorithm	101
Figure 69. Gradient descent objective function.	102

Key to Abbreviations

Pulsewidth Modulation.....	PWM
Sinusoidal Pulsewidth Modulation.....	SPWM
Random Pulsewidth Modulation.....	RPWM
Adaptive Random Pulsewidth Modulation.....	ARPWM
Modulation Index.....	MI
Direct Digital Synthesis.....	DDS
Fast Fourier Transform.....	FFT
Total Harmonic Distortion.....	THD
Electromagnetic Interference.....	EMI
Artificial Neural Network.....	ANN
Genetic Algorithm.....	GA
Field Programmable Gate Array.....	FPGA
Direct Memory Access.....	DMA
First-In, First-Out.....	FIFO

Introduction

Pulsewidth modulation schemes used in various electric power conversion applications can result in objectionable vibration and acoustic behavior resulting from electromagnetically induced mechanical response. The response will be more salient if the modulation carrier frequency coincides with a system mechanical resonance, especially if the carrier frequency is constant. Several random pulsewidth modulation (RPWM) techniques have been proposed over the last 20 years in order to reduce the prominence of the modulation over the 2 kHz to 5 kHz range. The area where these RPWM techniques are lacking arises when system response varies under different operating conditions, or when it is not known a priori. While effective at reducing objectionable content (either acoustic or electromagnetic) by spreading it over neighboring regions, they still do not address the robustness requirements of systems with varying responses. Additionally, the approach may actually increase the likelihood of coinciding with a system resonance, since the exciting frequency range is wider than in constant switching frequency techniques.

The proposed approach builds upon existing RPWM strategies by using an adaptive algorithm to position and shape the RPWM modulation frequency distribution based on system vibration feedback. Gradient descent is employed to vary the center frequency and distribution range of the randomized carrier frequency in response to change in the objective metric based on the vibration feedback.

Literature Survey

There are many published techniques for lessening objectionable audible sound and electromagnetic interference emanating from modulation schemes used in power electronics conversion applications. We will be concentrating on the frequency content in the audible range. Most reference papers identified in this area concentrate on some type of modulation technique, perhaps because these are most generically applied and usually do not require additional components or significantly more resources for the corresponding software. Many modulation techniques are available for diminishing objectionable consequences, including randomized modulation (RPWM) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], pulse skip modulation [11], slope modulation [12, 13], and sinusoidal modulation (with harmonic injection) [14].

The main idea behind these approaches is to shape the distribution of the modulation frequency in order to distribute energy to surrounding bands that could otherwise potentially coincide with resonant peaks or be sufficiently prominent to induce objectionable vibration on their own.

Other mitigation approaches include optimal switching pattern searches [15, 16], reference injection [17], ultra/hyper-sonic carrier frequencies [18], power electronics topology or component selection [19, 20, 21], and mechanical design (damping); these approaches are not as favorable as the modulation techniques, because they are more limited in existing applicable

systems and they may increase cost due to additional mechanical or electrical components or increased software complexity.

Non-RPWM Modulation Techniques

Sinusoidal modulation (Figure 1), typically implemented in combination with harmonic injection [14], works well for reducing the peakedness of the modulation frequency, however there is a chance that a system resonance could be excited by the frequency at which the carrier frequency itself is modulating.

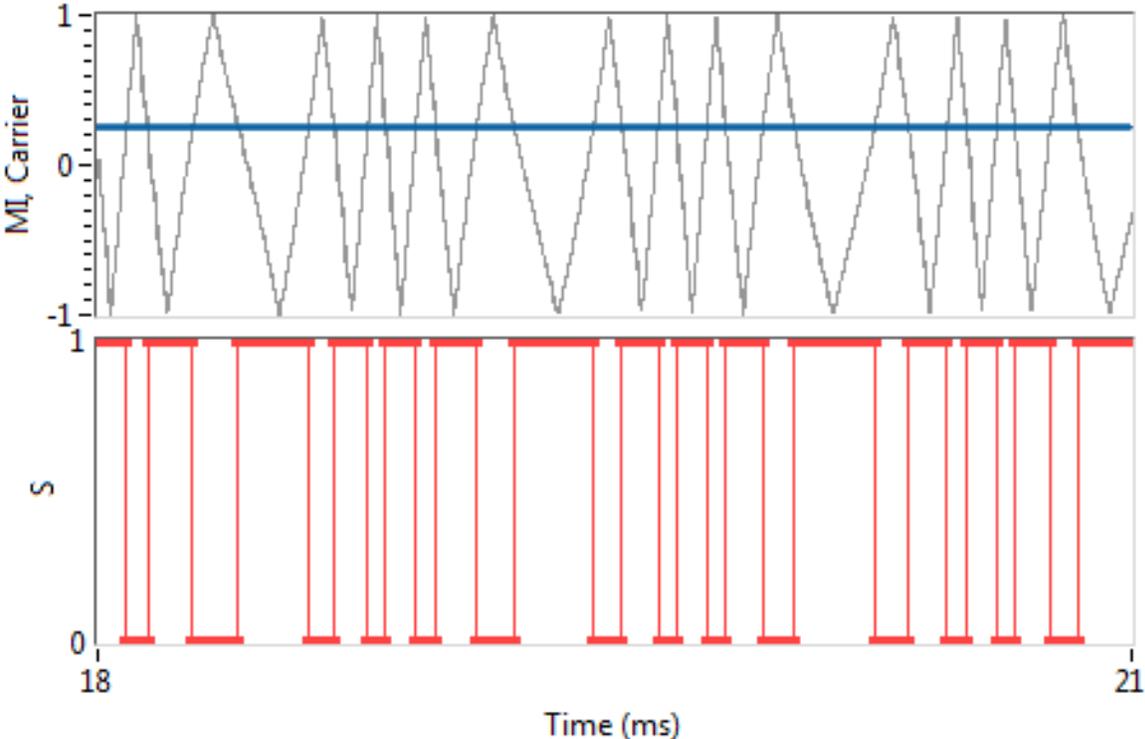


Figure 1. SPWM with a sinusoidally varying carrier frequency. For interpretation of the references to color in this and all other figures, the reader is referred to the electronic version of this thesis.

Pulse skip modulation [11] can also reduce prominent frequency content at the modulation frequency, however it may only be effective under light load conditions as it is based on combining short duration pulses at a high frequency into longer pulses modulated a lower equivalent frequency.

With slope modulation [12, 13], the carrier frequency is a function of the function of slope of the reference signal (Figure 2). This is typically employed in trapezoidal/six-step applications. In [12] a sinusoidal carrier signal is also proposed.

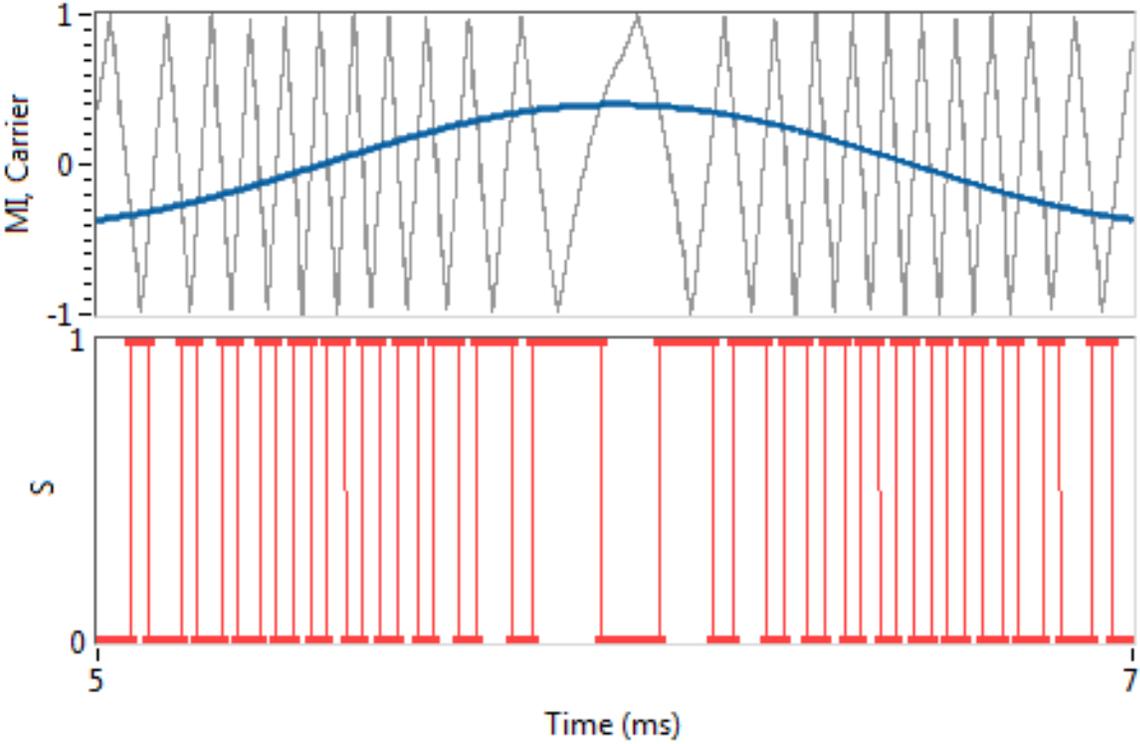


Figure 2. SPWM with carrier function whose frequency is a function of the slope of the reference signal. Although typically employed in trapezoidal/6-step controllers, here the concept is portrayed in an SPWM context.

RPWM Techniques

RPWM has three main variations [3] (Figure 3): random carrier frequency, random switching scheme (in which the carrier function is random), and random pulse position (lead-lag). There are also approaches that are hybrids of the three variations.

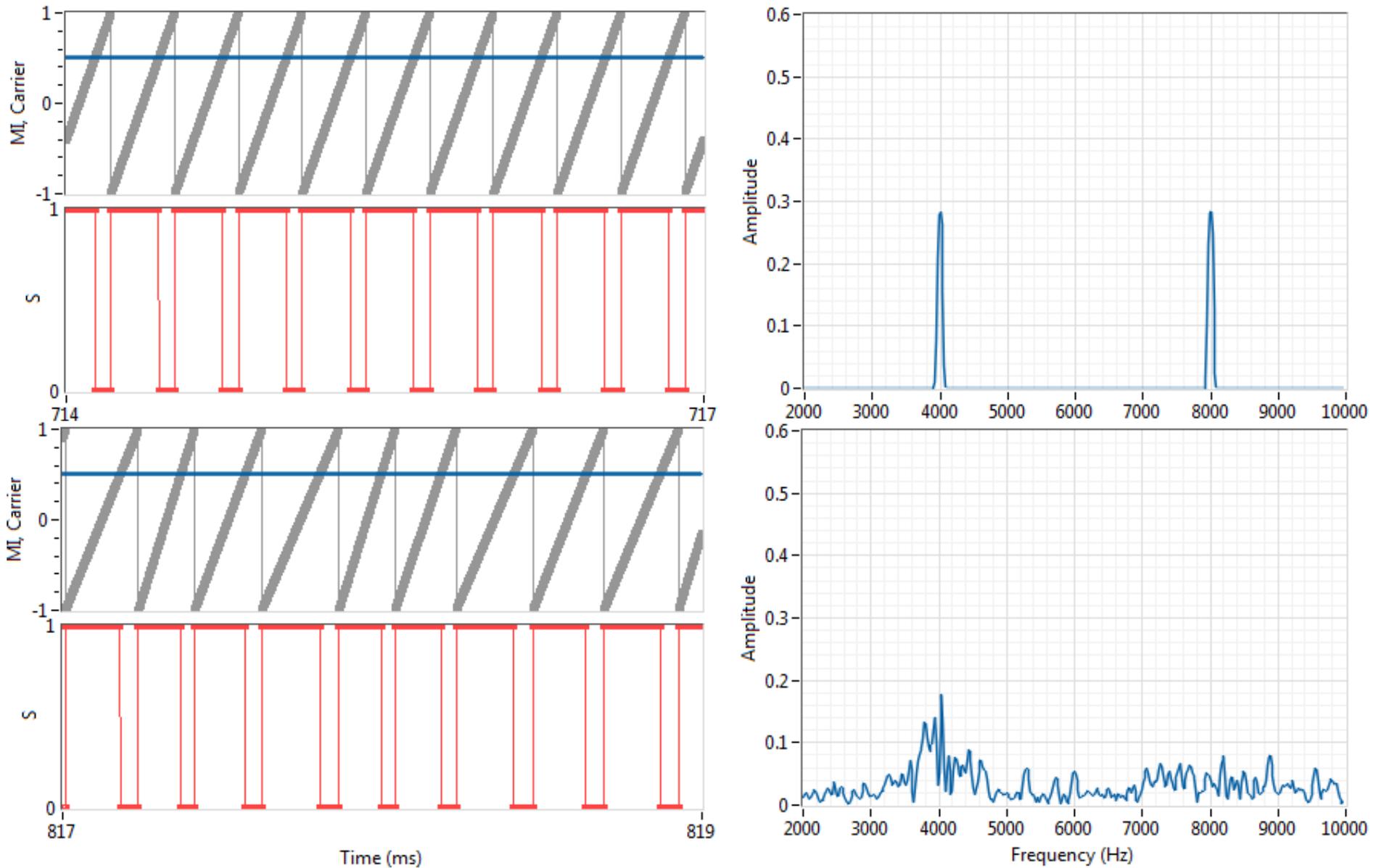
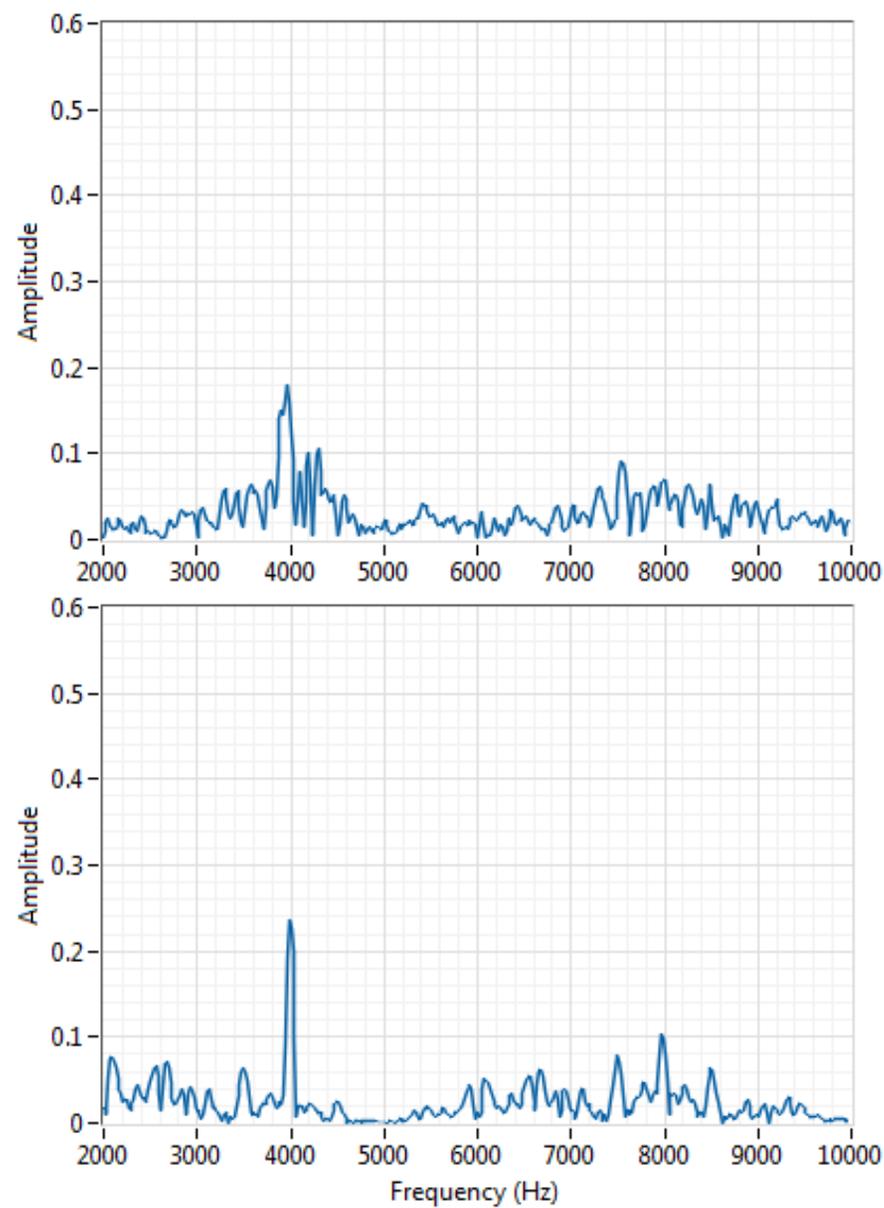
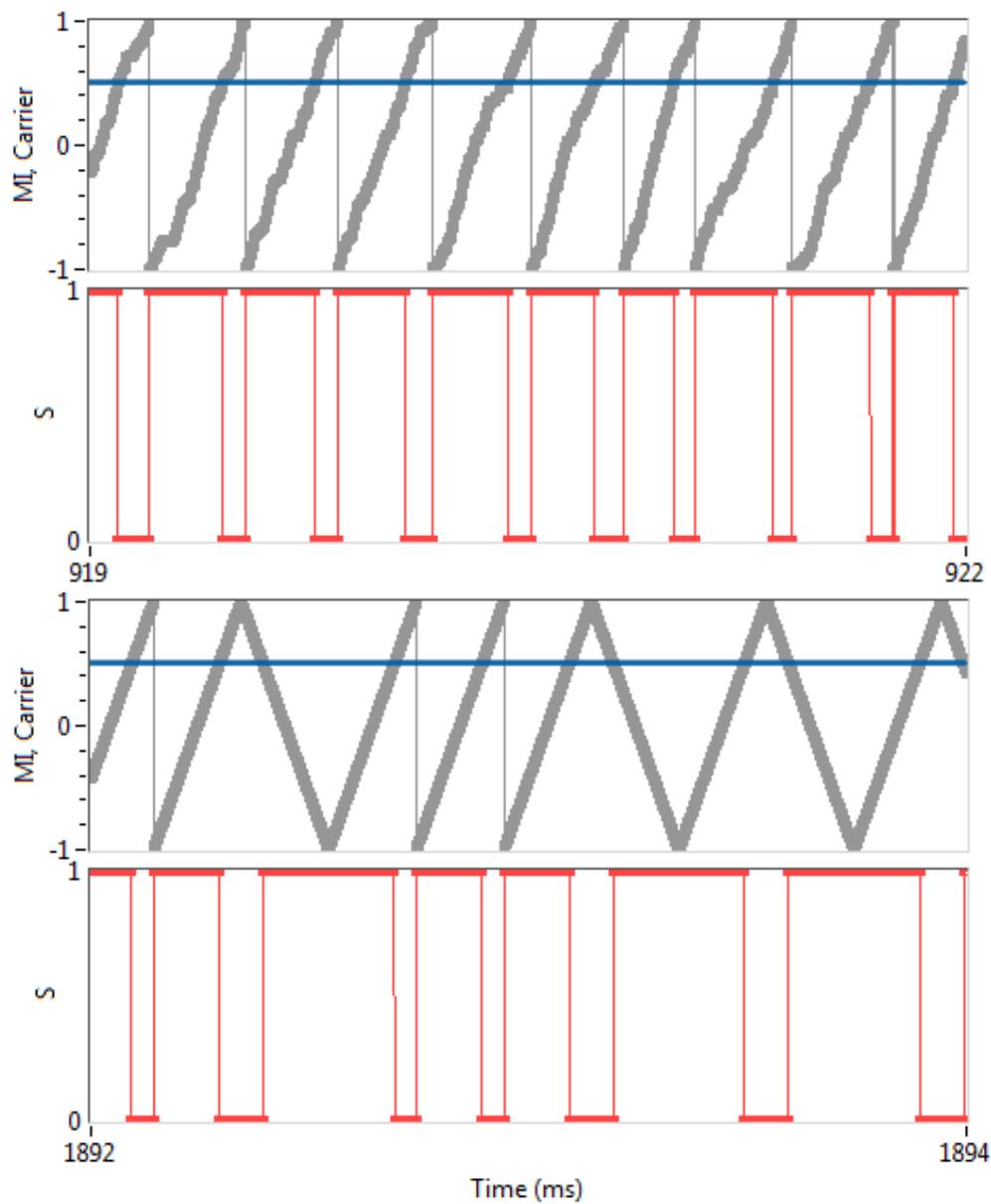


Figure 3. MI, Carrier, Switching Command S (left column), and PSD of S (right column) for SPWM (top), RPWM with random carrier frequency (2nd from top), RPWM with random carrier function (3rd from top), RPWM with random pulse position (bottom).

Figure 3 (cont'd)



Note that in the case of random pulse position, more variations are available than portrayed in Figure 3; the pulse could be arbitrarily located within the switching cycle, rather than at one side or the other.

In order to realize the random distribution, there are several approaches and distributions, including band limited white noise, quasi-random/M-sequence [22], Markov chains [5], grey-noise [10], and limited pool [8].

There are potential issues with employing RPWM approaches in an unidentified or changing system. If the modulation frequency distribution is static, there is susceptibility to objectionable mechanical response when the distribution resides in range of any resonances that appear under certain operating conditions. There is also an increased chance of exciting a system resonant frequency due to wider modulation frequency distribution.

Previous Adaptive Approaches Relevant to this Topic

In [23], a gradient descent approach is used to maximize efficiency by finding the optimal switching frequency for the given system state. This approach utilizes a very similar strategy as proposed here, however their objective is to maximize efficiency, rather than minimize objectionable system vibration. Additionally, only the frequency is adjusted, not any other characteristics related to the distribution of the frequency. In [24] acoustic output is measured and used as feedback, however only the volts/Hz constant is varied based on this feedback; this does not address objectionable mechanical response excited by modulation frequencies. If you combine the concepts introduced in these references, along with the concepts introduced in the RPWM references, you can arrive at the proposed approach in this thesis.

Proposed Technique

Since the system vibration response to the exciting voltage will be dependent on the state of the system (over varying angular velocities and loads, in the case of motors) vibration feedback will be employed to adaptively shape the exciting frequency content. Unlike typical RPWM, which may actually have an increased potential of exciting a system resonance due to the relatively wide excitation range [24], this approach will include adaptive control over the center frequency and range of the distribution.

If the system exhibits objectionable response to particular frequency ranges, and if this response may vary over different operating conditions, it is important to be able to relocate the majority of the exciting energy outside of any of these potentially changing natural frequency ranges. In order to do this, the designed spectral content will need a sufficiently wide range of allowable average switching frequencies, along with the ability to vary the span of distribution. Prior to arriving at the straightforward vibration feedback gradient descent algorithm with modulation frequency distribution center frequency and range (assuming a uniform distribution) as the independent variables, other configurations were considered. Other considered variations of the proposed gradient descent approach included adjustment of arbitrary probability density functions or more complex distributions to be described by additional higher order statistical parameters such as kurtosis and skewness, but both were forgone due to the simplicity and effectiveness of variable center frequency/range approach. Other adaptive approaches, alternative to gradient descent, were also considered in order to accommodate an arbitrary mechanical system response, and even a possibly varying response.

One concept was to use an artificial neural network (ANN) in order to realize a system identification of mechanical system response at different frequency ranges of interest. The identified responses could then be used to determine the best carrier frequency distribution to use for the current conditions. This approach seemed to be unnecessarily complex, and it seemed more appropriate and direct to just dynamically vary the carrier frequency distribution based on actual measured system response.

Another idea was to employ a genetic algorithm (GA) or particle swarm optimization (PSO) in order to search for a good carrier frequency distribution on-the-fly, however with these approaches, it was more likely that objectionable resonant peaks would be encountered and excited during the search; more frequently and possibly more severely than a hill-climbing approach (even though these types of searches would be less likely to arrive at a local minimum than an ill configured hill climbing approach).

Finally it was decided that a basic gradient descent hill-climbing technique would actually be computationally efficient, easy to implement, and effective. The key to this approach would be properly configuring the learning rate such that the descent would not be too aggressive - or not aggressive enough to the point is overly susceptible getting trapped in local minima occurring in the search space (where the profile of the search space is a function of the signature of the mechanical response spectrum). Additionally, the signature of the frequency domain spectrum is not expected to have very many deceptive local minima, which could be detrimental to a search technique that is essentially looking for local minima.

Gradient Descent

In order to implement a gradient descent algorithm when the value of the objective function is unknown throughout the search space, we will depend on our discretized feedback to measure changes in the objective search space as we traverse towards desirable valleys. Additionally, since our objection function is dependent on two variables (in this case, modulation frequency distribution center frequency and range), we cannot calculate partial derivatives with respect to the individual variables (Equation 1), since we will not know the contribution of each independent variable - since we don't have knowledge of the changing search space.

$$x_{n+1} = x_n - \gamma \nabla F(x), \quad x = \begin{bmatrix} \omega_c \\ r \end{bmatrix} \quad (1)$$

Where the x vector represents the state of our two independent variables, ω_c represents the center frequency of our modulation frequency distribution, r represents the range of the modulation frequency distribution, γ represents our instantaneous learning rate, and $F(x)$ represents the value of our objective function.

In order to overcome this, we will alternate between the independent variables at some interval, essentially creating two separate gradient descent searches. In this sense we do not need to know the function that defines the signature of the search space; we only need to calculate the measured change in our objective function and we know that this change is only with respect to the single variable that we have incremented. While the center frequency of the modulation frequency distribution is held constant, the gradient descent algorithm will be applied to the distribution range:

$$r_{n+1} = \begin{cases} r_n - \gamma \frac{\Delta F(r_n)}{\Delta r_n}, & \text{if adjusting distribution range} \\ r_n, & \text{if adjusting distribution center frequency} \end{cases} \quad (2)$$

$$\Delta F(r_n) = F(r_n) - F(r_{n-1}) \quad (3)$$

$$\Delta r_n = r_n - r_{n-1} \quad (4)$$

Similarly, while the distribution range is held constant, the gradient descent algorithm will be applied to the distribution center frequency.

$$\omega_{c,n+1} = \begin{cases} \omega_{c,n} - \gamma \frac{\Delta F(\omega_{c,n})}{\Delta \omega_{c,n}}, & \text{if adjusting distribution center frequency} \\ \omega_{c,n}, & \text{if adjusting distribution range} \end{cases} \quad (5)$$

$$\Delta F(\omega_{c,n}) = F(\omega_{c,n}) - F(\omega_{c,n-1}) \quad (6)$$

$$\Delta \omega_{c,n} = \omega_{c,n} - \omega_{c,n-1} \quad (7)$$

It would also be possible to expand this concept to further independent variables, if it was desirable to describe the distribution with additional parameters. The algorithm would need to cycle round-robin style through more parameters, affecting the temporal response to a possibly changing search space (response). In order to counter this, the analog sampling rate and FFT frame size of the system could be adjusted to realize the desired behavior. Things such as variable learning rate could also be employed, so a higher rate is applied in higher vibration response regions for faster convergence.

One objective function employed for the gradient descent algorithm involves calculation of the maximum response occurring within the exciting range of the modulation frequency (center frequency +/- half of the range), for both first and second order.

$$\begin{aligned} \sqrt{F(\omega_c, r)} = & \max[X(\omega)], \omega \in [\omega_c - \frac{r}{2}, \omega_c + \frac{r}{2}] \\ & + \max[X(\omega)], \omega \in [2\omega_c - r, 2\omega_c + r] \end{aligned} \quad (8)$$

Where $X(\omega)$ is the vibration response amplitude (rms) at frequency ω .

It is important to include second order response in the function because this region is often objectionable too. In some cases, other contributions could interfere with the vibration feedback, such as orders of motor fundamental rotation, but this is not a concern since in our area of concentration, the vibration excited by the modulation will be prominent, and it will be the dominant contributor to the frequency content. Another detail that requires attention is the type of window applied to each frame of data before the frequency domain transformation is performed. Here it is appropriate to apply a flattop window in order to get a good representation of the amplitude of the vibration occurring at the frequency ranges of interest.

An alternative objective function was employed while incorporating a variable learning rate, since our learning rate selection algorithm required a less noisy objective function than our constant learning rate configuration.

$$F(\omega_c, r) = \frac{\int_{\omega_c - \frac{r}{2}}^{\omega_c + \frac{r}{2}} X(\omega) d\omega}{r} + \frac{\int_{2\omega_c - r}^{2\omega_c + r} X(\omega) d\omega}{2r} \quad (9)$$

First, instead of returning the peak vibration level over the range of interest, the average vibration level over that range is calculated. Additionally, we are no longer taking the square of the resulting value, since this would further exaggerate any measurement noise (and since the learning rate is variable, we no longer need to exaggerate the search space to accommodate an unchanging learning rate).

Convergence and Stability and Variable Learning Rate

In order to define a strategy to implement the gradient descent algorithm with a variable learning rate, it is important to understand the behavior of the system under the conditions that may be encountered.

We begin with our entire search space (Figure 4), composed of the value of our objective function at each modulation frequency distribution center frequency and range combination.

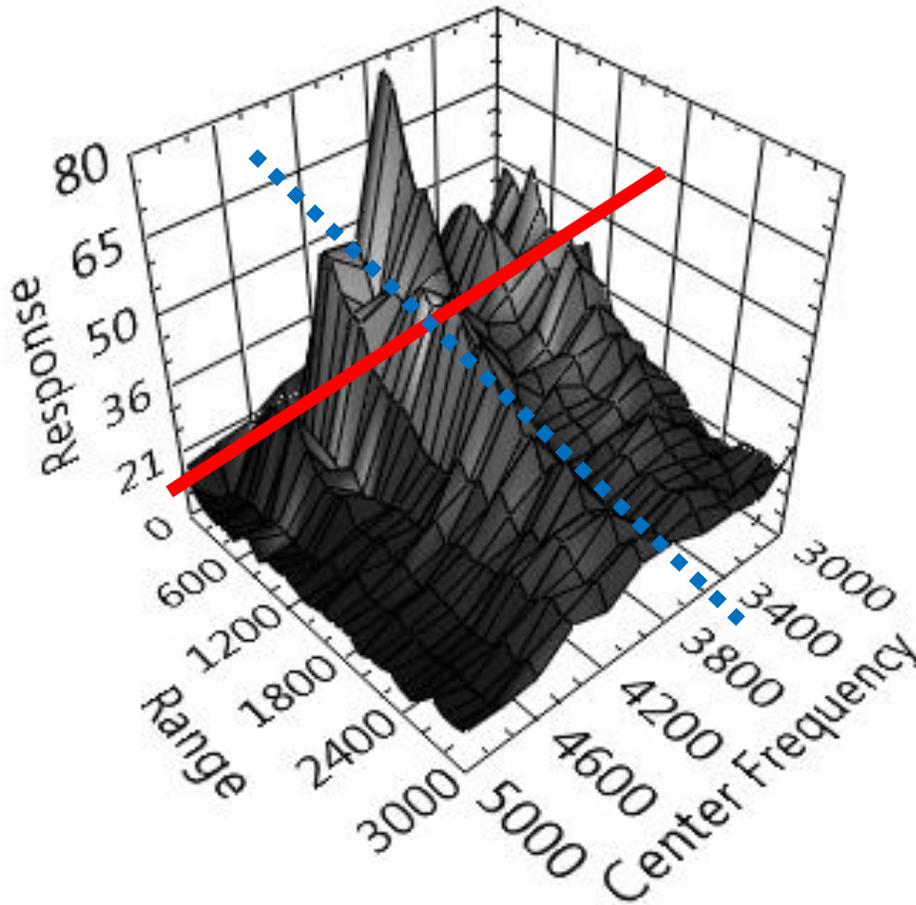


Figure 4. Typical search space with constant range and center frequency cross sections indicated with lines. Signatures from these cross sections will be used for our analysis.

Since we are essentially performing two separate gradient descent operations, each having one independent variable, we only need to analyze a single slice of our search space, and then generalize our findings to the entire search space. Figure 5 and Figure 6 plot the signature of the search space at constant center frequency and constant range, respectively.

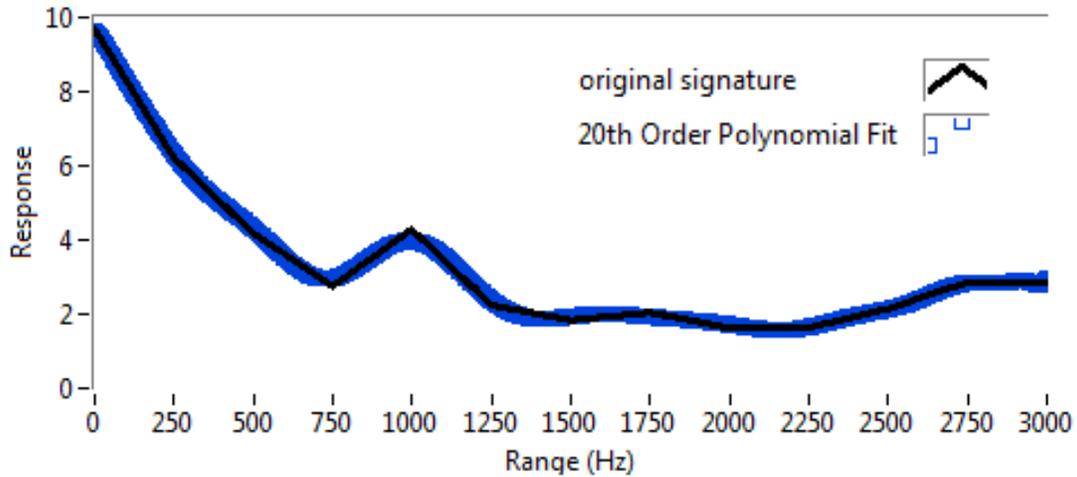


Figure 5. A constant center frequency slice signature from the search space

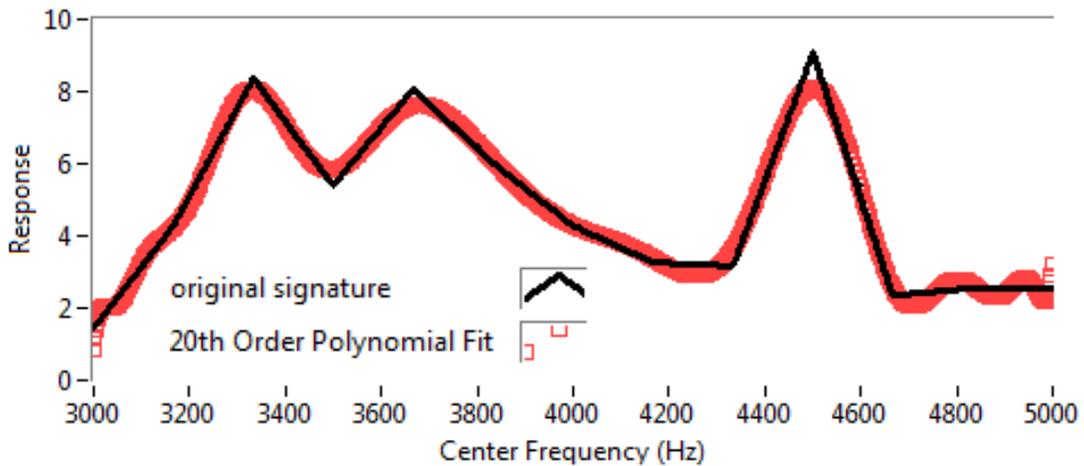


Figure 6. A constant range slice signature from the search space

To simplify the analysis, we will make some assumptions. In order to be able to perform the convergence analysis, we will assume that the system response and objective function can be approximated by a polynomial of degree n . Later, we can vary the polynomial over the expected range and signatures of system response that may be encountered. This way we can determine behavior for a given system response and learning rate, and adjust the learning rate to achieve the fastest possible response without introducing instability.

In order for this analysis to be applied to the actual system, the response will need to be appropriately calculated and possibly averaged in order to minimize the effects of unexpected noise. It is assumed that we will find that the learning rate will be variable, and that we will have a separate learning rate for the center frequency and range gradient descent algorithms. Also, the learning rates will be a function of a few metrics based on encountered conditions, which will be used to estimate the shape of the search space, which will vary over time.

We want to express the convergence and stability as a function of the learning rate and the polynomial fit to the system. Given this information, we can either identify the system by fitting the measured changes in independent variable and response objective function to a polynomial and then selecting the most appropriate learning rate, or we can just 'roughly' calculate our learning rate based on the largest changes encountered. To begin, we will define the system block diagram for context, but we will concentrate on the convergence of the algorithm.

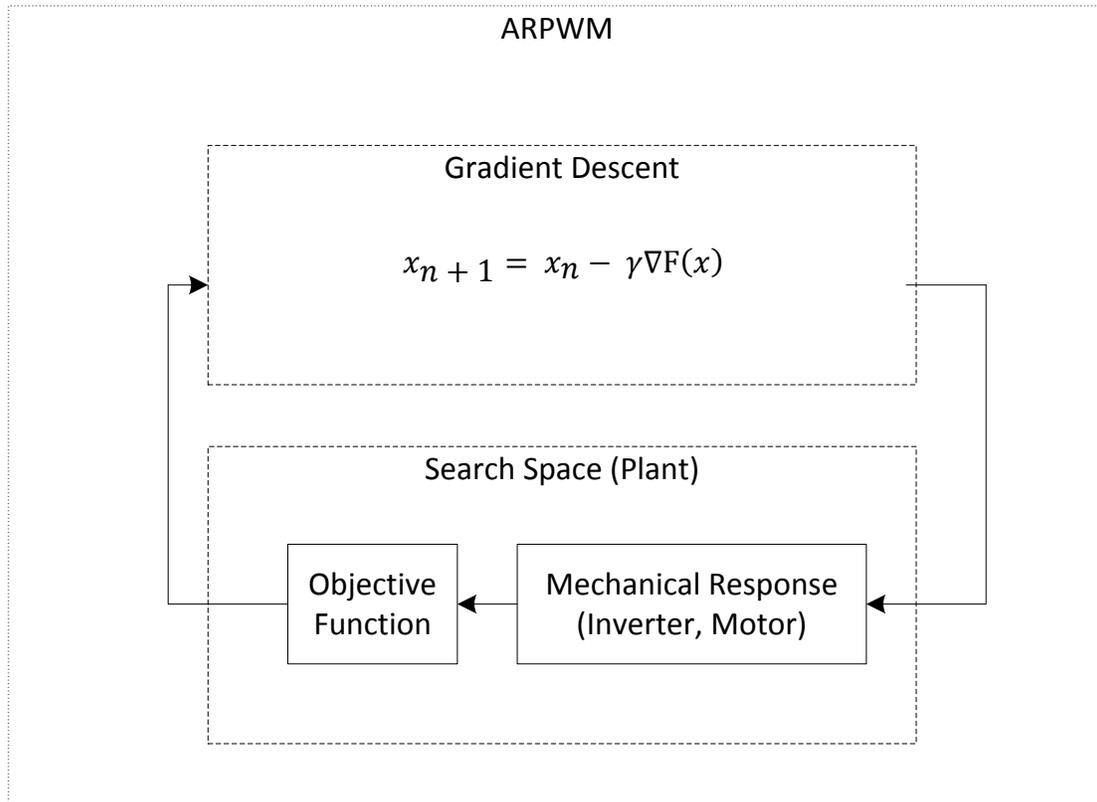


Figure 7. Block diagram of ARPWM with gradient descent

Initially, attempts were made to solve the recurrence formula for an explicit expression of the independent variable at any step, but this was not only difficult, but unnecessary. We can either analyze the system graphically by iterating with the recurrence relationship, or we can use an analytical approach to design our variable learning rate algorithm (we will do both here). The following expressions describe the gradient descent algorithm and the polynomial approximation of the objective function F . If we consider one of our independent variables, the modulation frequency distribution center frequency ω_c , the gradient descent equation can be used to describe the rate of change of the independent variable as a function of the gradient

$\frac{\dot{F}}{\dot{\omega}_c}$ and learning rate γ :

$$\dot{\omega}_c = \omega_c - \gamma \frac{\dot{F}}{\dot{\omega}_c} \quad (10)$$

Since we are approximating the objective function result with a n^{th} order polynomial, we get:

$$F = \alpha_0 + \alpha_1 \omega_c + \alpha_2 \omega_c^2 + \alpha_3 \omega_c^3 + \dots + \alpha_n \omega_c^n \quad (11)$$

Or in summation form:

$$F = \sum_{j=0}^n \alpha_j \omega_c^j \quad (12)$$

The derivative is easily defined as:

$$\frac{dF}{d\omega_c} = \alpha_1 + 2 \alpha_2 \omega_c + 3 \alpha_3 \omega_c^2 + \dots + n \alpha_n \omega_c^{n-1} \quad (13)$$

Or in summation form:

$$\frac{dF}{d\omega_c} = \sum_{j=1}^n j \alpha_j \omega_c^{j-1} \quad (14)$$

We can restate the gradient descent equation:

$$\omega_{c,k+1} = \omega_{c,k} - \gamma \nabla F(\omega_{c,k}) \quad (15)$$

And then substitute the derivative of our polynomial approximation for the gradient:

$$\omega_{c,k+1} = \omega_{c,k} - \gamma \sum_{j=1}^n j \alpha_j \omega_{c,k}^{j-1} \quad (16)$$

With this relationship, we can construct the graphics and analyze the system behavior. Note that in the following analysis, the gradient descent independent variable is scaled to the

interval [0,1], so any learning rates, derivatives, or second derivatives are calculated over this interval; the values are scaled up to Hz units just for the generation of the graphics (this is only mentioned to explain why the plots of the derivatives are scaled versions of the derivatives had they been calculated on the functions in engineering units rather than normalized units).

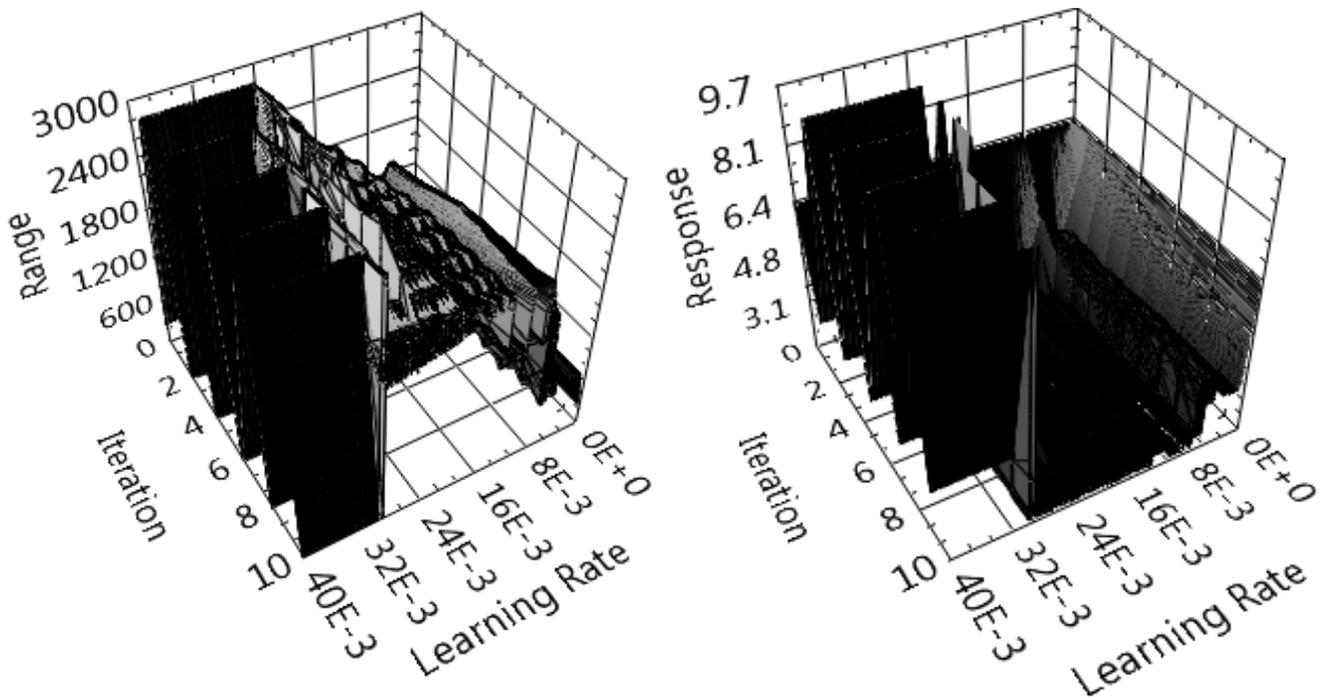
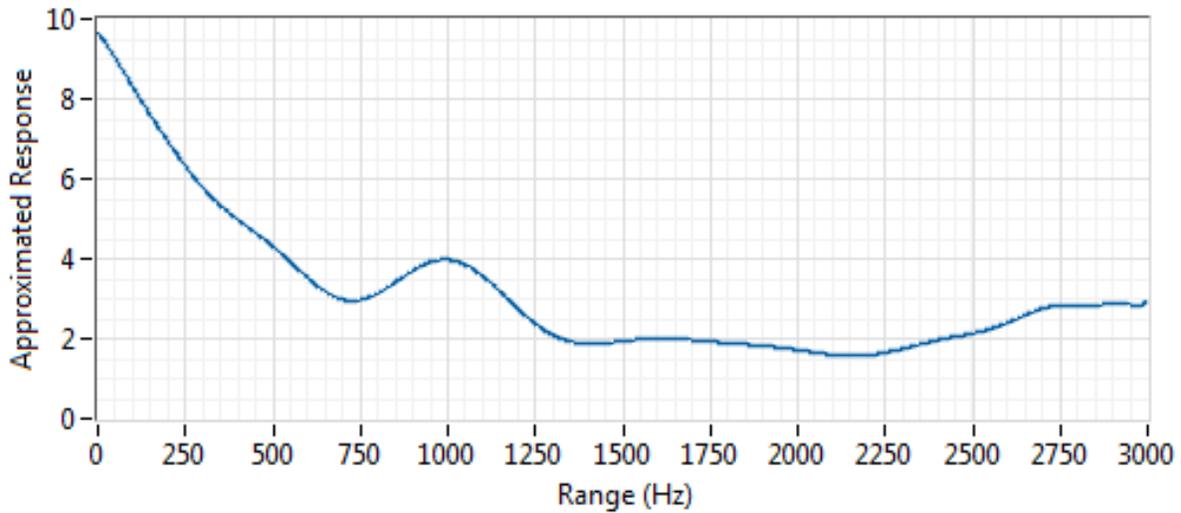


Figure 8. Response and Range versus learning rate versus iteration number (starting at a high-response location) for the response signature as range is varied over a constant distribution center frequency.

From Figure 8, we can see that for the given initial conditions, learning rates less than 0.005 get stuck in local minima while learning rates greater than 0.03 are unstable.

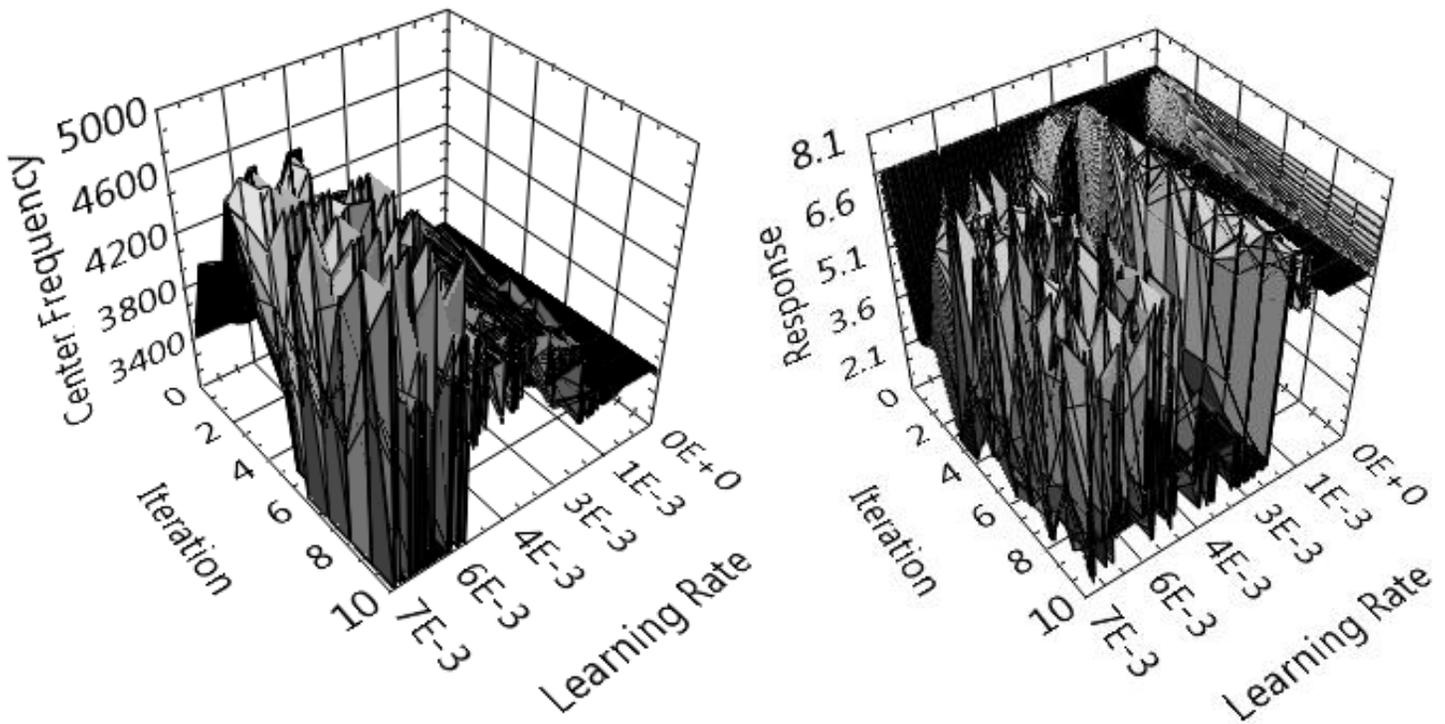
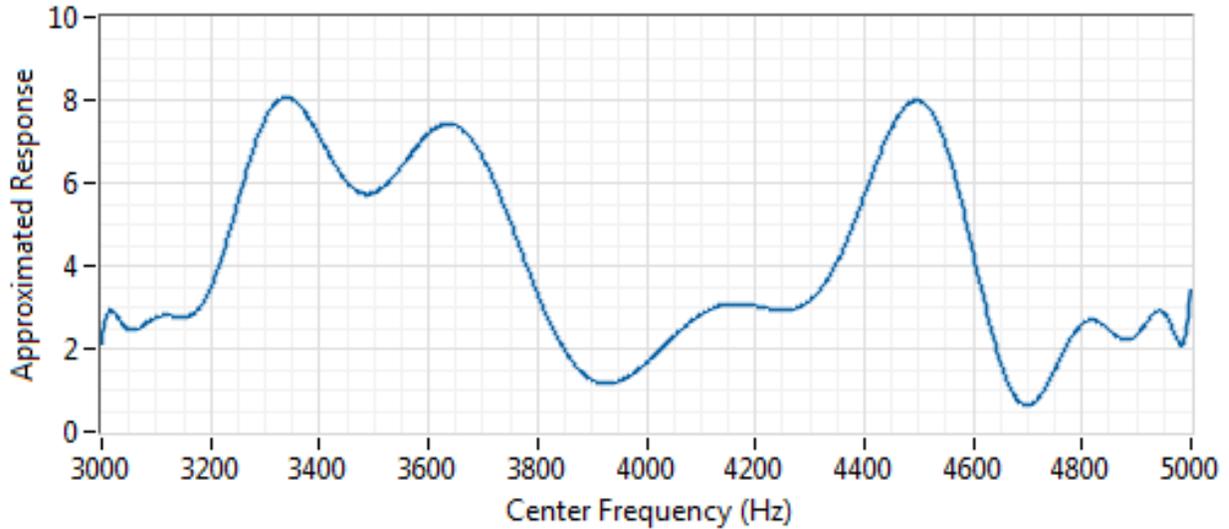


Figure 9. Response and Range versus learning rate versus iteration number (starting at a high-response location) for the response signature as center frequency is varied over a constant distribution range (starting somewhere in the local minimum at 3400Hz).

As seen in Figure 9, learning rates greater than 0.005 are unstable for this signature. While with a learning rate below 0.002 behavior is very stable, learning rate needs to be greater than 0.003 in order to jump out of the local minimum.

Several generalizations can be formed by analyzing the graphics. The selection of learning rate is highly dependent on the signature of the search space. A higher learning rate is sometimes required to overcome local minimums before reaching the global minimum. A variable learning rate is desirable to realize stability near global minimums and capability to escape local minima at relatively high response levels. A separate acceptable range of learning rates should be used for the two independent variables.

While it is easy to make these observations and generalizations graphically, we still would gain more insight from an analytical study. We can use Equation 17 in order to define the maximum stable learning rate for arbitrary response signatures.

For a non-linear first-order recurrence (first order in the sense that the next value is only a function of the previous value),

$$x_n = f(x_{n-1}) \quad (17)$$

Local stability is realized if

$$|f'(x^*)| < 1 \quad (18)$$

where x^* is a nearby local minimum [25].

Furthermore, if the system is oscillating over multiple steps, the convergence will occur if the composite function $g(x)$ of those individual steps ($f(x)$) is stable.

$$g(x) := f \circ f \circ \dots \circ f(x) \quad (19)$$

$$|g'(x^*)| < 1 \quad (20)$$

Recalling the value of our objective function (Equation 11) and its first derivative (Equation 13), we can similarly define the second derivative:

$$\frac{d^2F}{d\omega_c^2} = 2 \alpha_2 + 6 \alpha_3 \omega_c + 12 \alpha_4 \omega_c^2 + \dots + n(n-1) \alpha_n \omega_c^{n-2} \quad (21)$$

Or in summation form:

$$\frac{d^2F}{d\omega_c^2} = \sum_{j=2}^n j(j-1) \alpha_j \omega_c^{j-2} \quad (22)$$

From Equation 18, 15, and 22, we know that convergence to a nearby local minimum is realized if

$$1 > \left| \frac{d[\omega_{c,k} - \gamma \nabla F(\omega_{c,k})]}{d\omega_{c,k}} \right| = \left| 1 - \gamma \sum_{j=2}^n j(j-1) \alpha_j \omega_c^{j-2} \right| \quad (23)$$

Which can be rewritten as

$$0 < \sum_{j=2}^n j(j-1) \alpha_j \omega_c^{j-2} < \frac{2}{\gamma} \quad (24)$$

From this relationship, we can determine stable learning rates for arbitrary response signatures (Figure 10 and Figure 11).

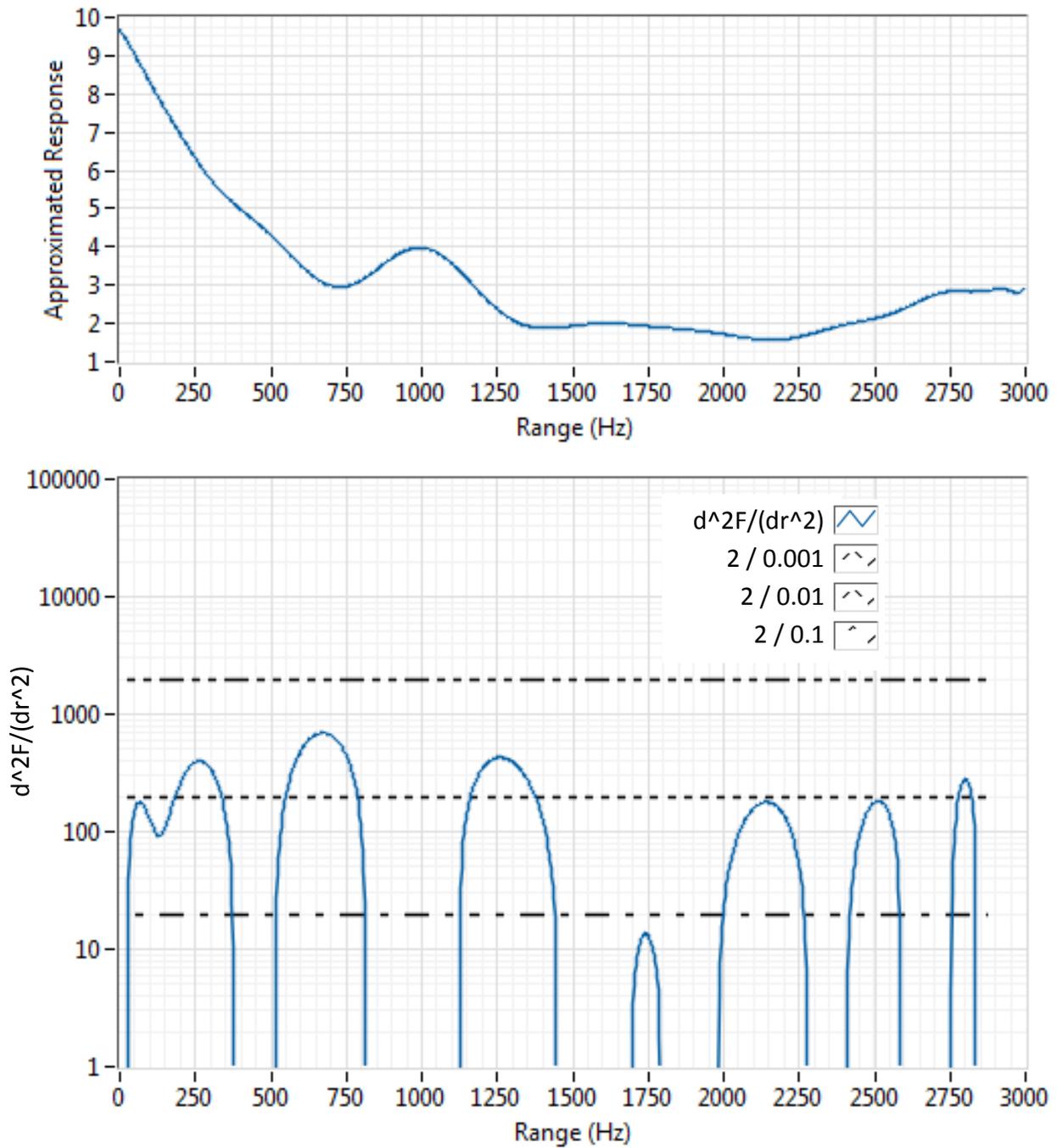


Figure 10. Analysis of local minima for typical response vs range for a constant center frequency slice

To converge into a local minimum for this type of signature (Figure 10), the learning rate must be between 0.01 and 0.0075.

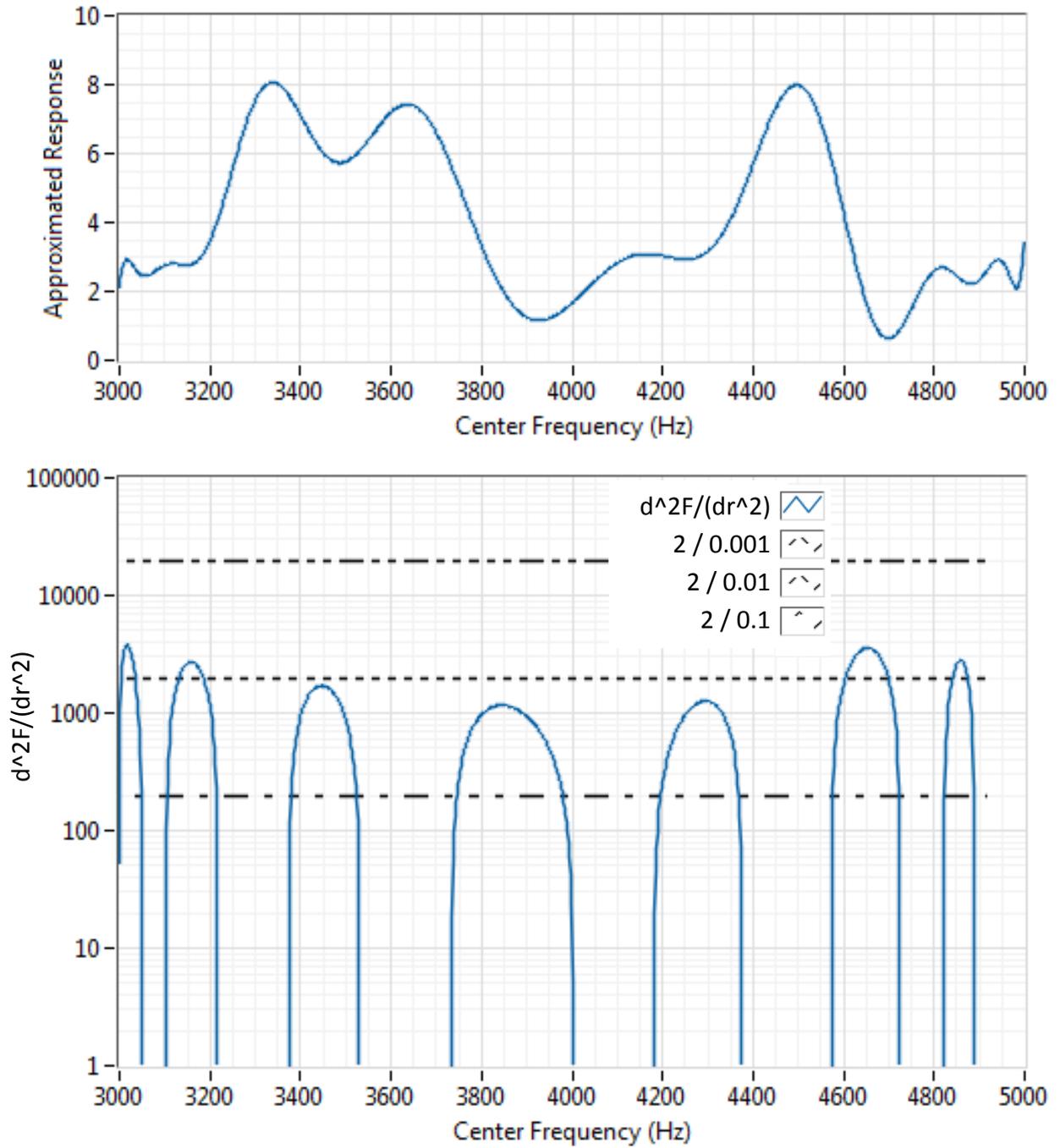


Figure 11. Analysis of local minima for typical response vs center frequency for a constant range slice

To converge into a local minimum for this type of signature (Figure 11), the learning rate must be between 0.001 and 0.0005 (about 10 times less than for the signature of the range).

Based on this information, an algorithm to implement a variable learning rate gradient descent has been designed such that 1) the peak learning rate Γ is a function the maximum encountered second derivative of the objective function with respect to the independent variable and 2) the instantaneous learning rate is a function of the peak learning rate and the ratio of the current objective value $F(\omega_c, r)$ to the maximum encountered objective F .

$$\gamma = \Gamma \sqrt{\frac{F(\omega_c, r)}{F}} \quad (25)$$

$$F = \max(F(t)) \quad (26)$$

$$\Gamma_{\omega_c} = \frac{2}{\max\left[\frac{d^2 F(\omega_c)}{d\omega_c^2}\right]} \quad (27)$$

$$\Gamma_r = \frac{2}{\max\left[\frac{d^2 F(r)}{dr^2}\right]} \quad (28)$$

The algorithm maintains running maximum values for the objective metric (Equation 26) and second derivative (Equation 27, Equation 28). This square root of the current objective value to maximum objective value ratio is multiplied by the maximum learning rate to obtain the instantaneous learning rate (Equation 25). The square root is used in case if one of the two

independent variables converges faster than the other, so the learning rate will still be high enough for the other to converge in a timely manner. With this basic approach, convergence rate will increase proportionally with system response (objective metric value), but will back off when nearing a global minimum. Additionally, the peak learning rate is calculated, rather than estimated through offline analysis or trial-and-error.

Cost Considerations

In order to eliminate the need for an accelerometer, one approach could be to train the controller at an end-of-line station (memorize distributions to use in a 'production line training' procedure), however this would only be applicable to a limited range of applications, since in many cases the system response is dependent on any number of installation variables and environmental conditions. If production line training were desirable, for motors one could either statically map (or train an adaptive algorithm) to be able to store (or generalize) optimal modulation carrier distributions to be looked up (or calculated) based on the current state of the system. For DC-DC converters, the optimal distributions could be recalled based on current conditions such as input and output voltages and currents and modulation index.

Additionally, there are inexpensive piezoelectric transducers available that could be employed for vibration feedback for very low cost. A quick internet search for low quantity orders yields some models with 5 kHz bandwidth for less than \$1 USD [26] , while models with bandwidth up to 32 kHz [27] can cost closer to \$20 in low quantities. In high quantities and in the right applications, the additional cost would be acceptable.

Implementation

Figure 12 depicts the block diagram of the proposed system. For development purposes, the gradient descent is temporarily placed on the PC rather than on FPGA (Field Programmable Gate Array) to allow faster implementation of adjustments to the algorithm without the need to perform time consuming recompiles of the FPGA code.

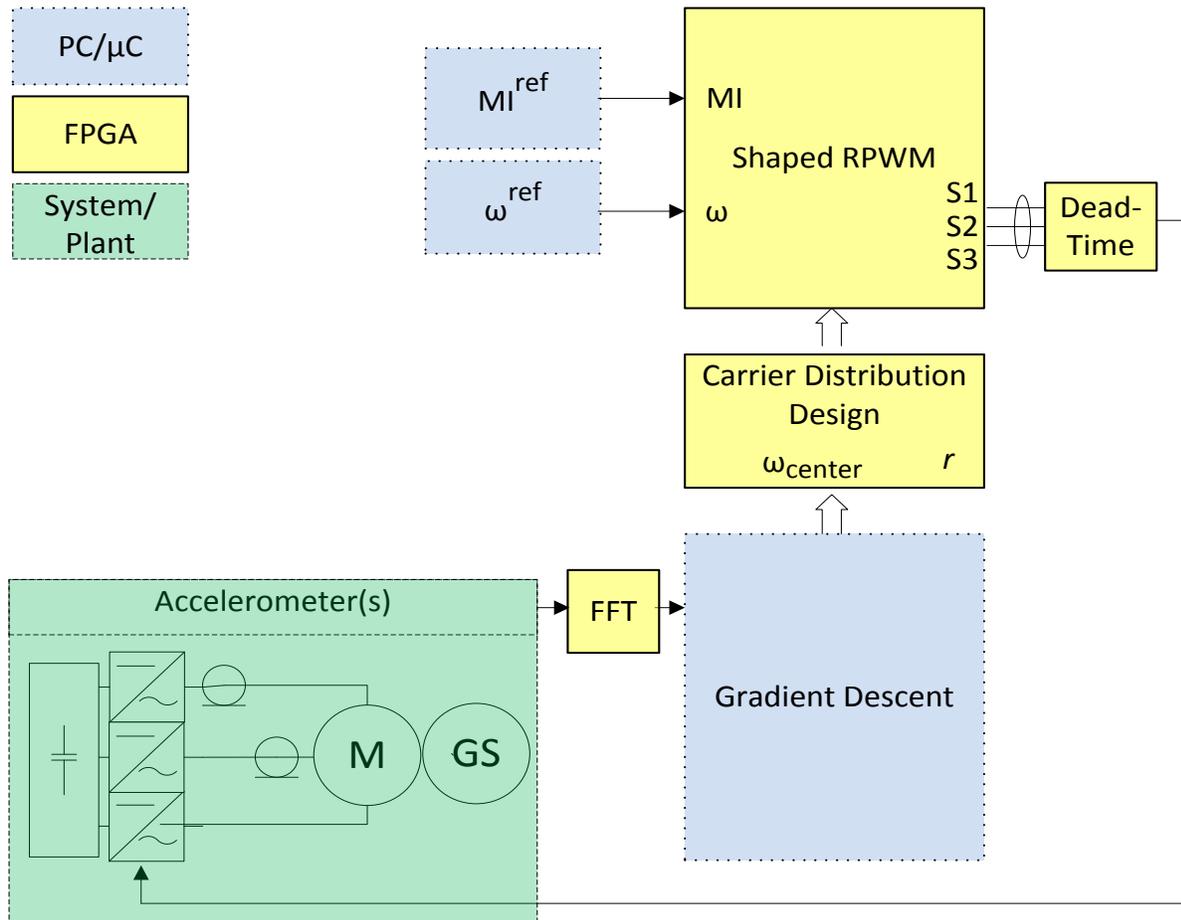


Figure 12. Adaptive RPWM (ARPWM) FPGA implementation

Software Overview

The FPGA and host code can be found in Appendix B. In this section an overview of the software will be provided. The FPGA roles consist of analog measurement and filtering, FFT

transformation of vibration data, generation of carrier frequencies based on the specified distributions (center frequency and range), and execution of RPWM (all of the deterministic and resource-intensive operations). The PC program provides an interface for manual adjustment of the peak modulation index and frequency of the reference signal. The ARPWM gradient descent was also implemented on the PC side – the one-sided spectrum is transferred from FPGA to the PC using a DMA (direct memory access) FIFO (first-in, first-out) queue. As the PC performs the ARPWM objective calculation and gradient descent iterations, it periodically returns new center frequency and range commands to FPGA so it begins to generate carrier frequencies that reflect the new distribution.

Design of Numerical Representation and Loop Rates

In digitally implemented RPWM applications, it is important to analyze and understand the effects of timing and numerical representation resolution on the control signals, since poor frequency resolution or large quantization error can severely reduce or completely eliminate the effectiveness of RPWM.

As depicted in Figure 13, the fidelity of the actual output switching commands to the intended output is affected with a digital implementation, but as long as the numerical representations and control loop times are selected appropriately, variation from the desired signal can be kept within an acceptable range. The granularity of the RPWM distribution is also affected, so the quantization errors and loop rates must be designed such that the range of the random distribution is much larger than any error – thereby ensuring that the random distributions are

actually distributed among a sufficient quantity of discrete frequencies and, more importantly, remain as symmetric as possible about the desired average frequency.

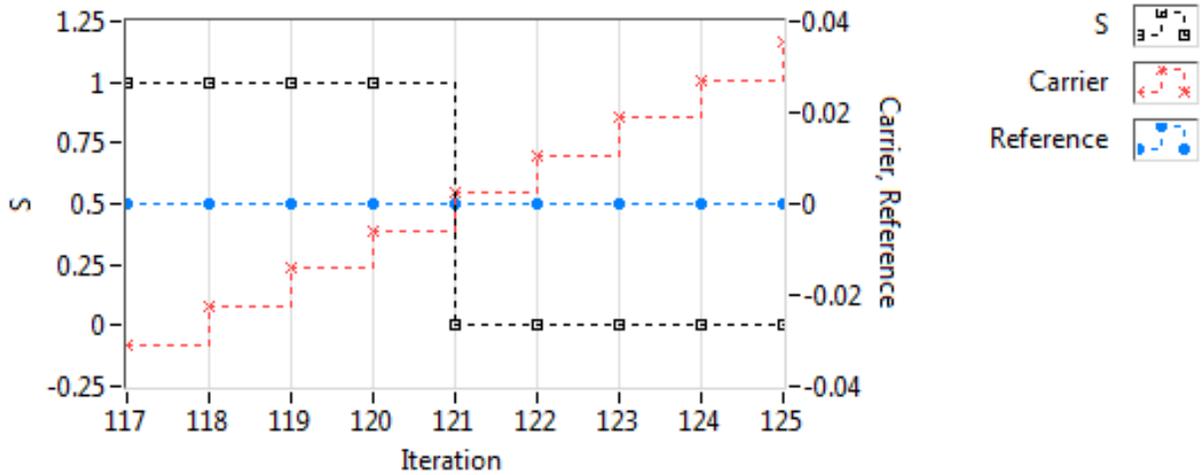


Figure 13. PWM switch command S , carrier, and reference signals versus control loop iteration with discrete implementation. The actual transition times are dependent on our numeric representations and timing resolutions.

In this case, our control loop rate is 4 MHz (250ns per iteration), and our sawtooth carrier and sinusoidal reference signals are represented with 32-bit signed integers. The 4 MHz loop rate allows for sufficient (but not ideal) behavior for this system, while the 32-bit signals offer more than sufficient resolution for a negligible quantization error. Further pipelining could be implemented to increase control loop rate at the expense of output control signal propagation delay; however for this application, the 250ns timing resolution will be sufficient.

The effect of the time and amplitude discretization of the carrier and reference signal on the desired output is dependent on the instantaneous modulation index (MI) and the switching

frequency. Figure 14 depicts experimental results obtained by sweeping the carrier frequency from 1 kHz to 10 kHz in 0.01Hz increments, while comparing the desired switching frequency with the reciprocal of the actual switching period and plotting the running unchanged output period length versus switching frequency.

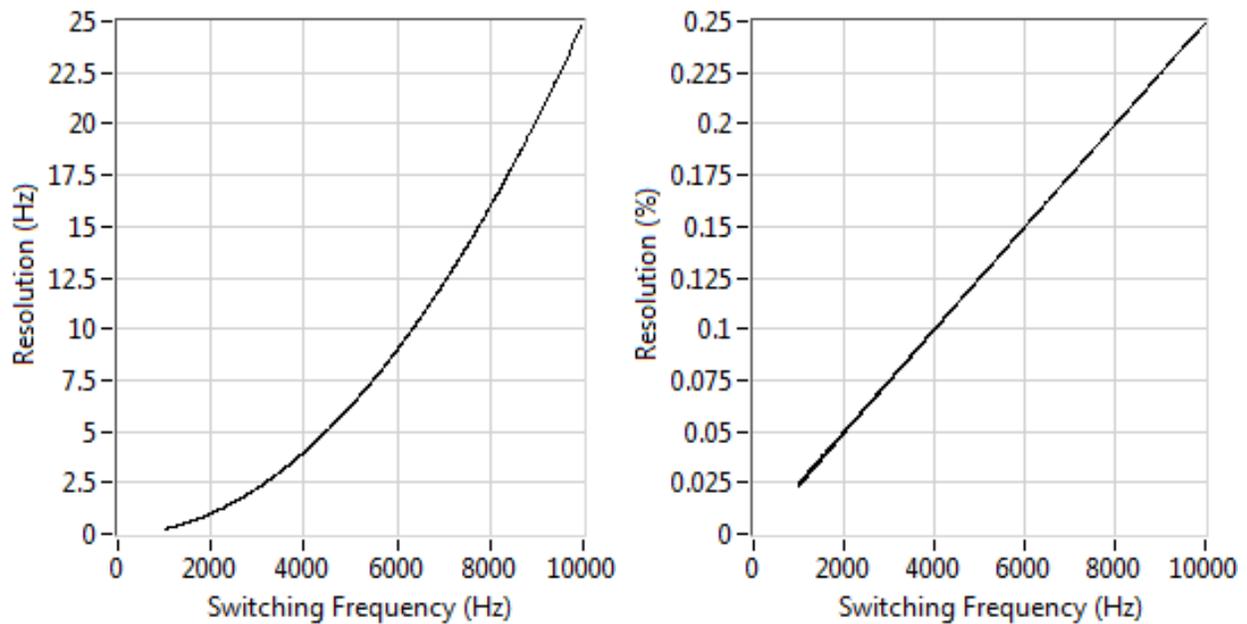


Figure 14. Experimental frequency resolution (or maximum error) versus switching frequency at 250ns control loop rate (left) and frequency resolution as a percentage of the switching frequency versus switching frequency (right). (when not constrained by insufficient numerical resolution of the carrier and reference signals)

Figure 14 plots the frequency resolution versus switching frequency, and the frequency resolution as a percentage (of the switching frequency) versus switching frequency. Since there are fewer samples per cycle at higher frequencies, the delta t becomes more significant as switching frequency increases. When expressed as a percentage of switching frequency, it becomes apparent that the frequency resolution worsens proportionally to the switching frequency (as the “control-loop-period-to-switching-period” ratio becomes larger).

Since the ratio of the control loop period to switching period indicates the quantity of possible discrete switching instances over a single switching cycle, the resolution can be expressed as a function of the control loop rate and switching frequency (when not constrained by carrier and reference signal numerical representation resolution):

$$\Delta\omega_{switching} = \frac{\left(\frac{1}{\omega_{control\ loop}}\right)}{\left(\frac{1}{\omega_{carrier}}\right)} * \omega_{carrier} = \frac{\omega_{carrier}^2}{\omega_{control\ loop}} \quad (29)$$

The implications of this relationship let us know that in order for our RPWM scheme to be effective, our allowable switching frequency distribution variation will need to be sufficiently large at higher switching frequencies to allow for proper redistribution of spectral content to neighboring spectral regions. For example, if our average switching frequency is 10 kHz, it would be desirable to have a random switching distribution spanning much more than the ~25 Hz resolution; otherwise RPWM will have little to no effect at all.

The effect of switching frequency on duty cycle is not as significant (for our purposes) as it was on the frequency resolution, although it is closely related and of the same order. It is evident that the error pattern of the duty cycle (Figure 15) is a function of the MI (reference signal), the frequency resolution (number of intervals in the switching cycle), and the phase relationship between the carrier and reference signals.

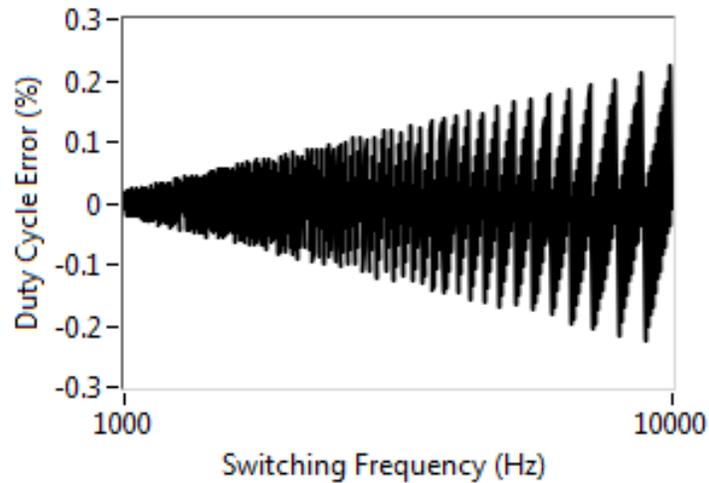


Figure 15. Duty cycle error (%) versus switching frequency when MI = 0.9.

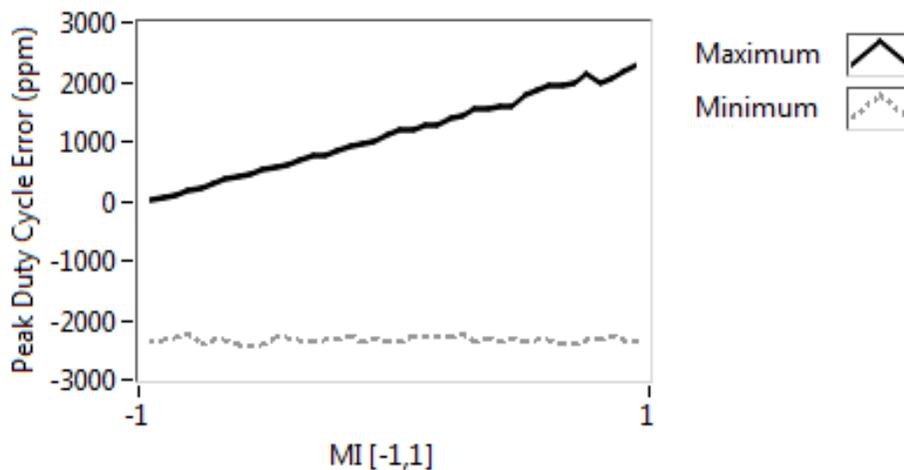


Figure 16. Peak duty cycle error (ppm) versus MI. At each MI, the frequency was varied from 1 kHz to 10 kHz and the maximum and minimum values were plotted. In this case the carrier frequency had no phase shift, so this is not representative of the worst case duty cycle error.

It is interesting how the error in Figure 16 tends towards the negative side. In order to better understand how the phase of the sawtooth effects the error in the duty cycle, Figure 17 plots the maximum and minimum peak duty cycle vs. carrier phase. This is important to understand, since we are randomly varying our carrier frequency, unlike in Figure 16, the phase relative to

the reference signal could be anywhere, so we can use the worst case duty cycle error as our design metric.

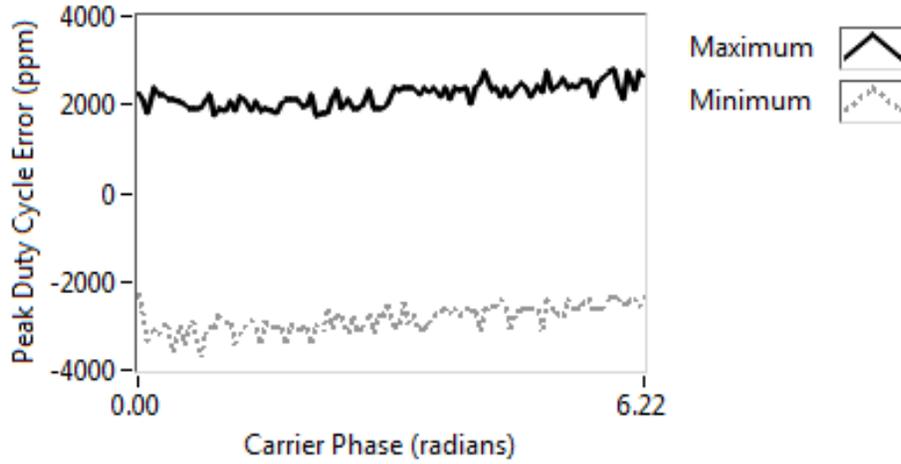


Figure 17. Peak duty cycle error versus carrier phase (evaluated with at 100 equally divided intervals). This just gives a rough idea of the error, since we skipped over several phase angles and several MI values in order to generate this graphic in a timely manner.

To simplify this evaluation, the relationship between the control loop rate, the switching frequency, and the worst case duty cycle error can be expressed as:

$$|D_{actual} - D_{Desired}| < 2 * \left(\frac{\omega_{carrier}}{\omega_{control\ loop}} \right) \quad (30)$$

Where the error is expressed as the difference between the actual and desired duty cycle. The reason for doubling the product of the control loop and switching periods is because either the leading or trailing transition time could be off from the desired transition time. This calculation is consistent with our experimental data, since for a 4 MHz control loop and a 10 kHz switching frequency, we end up with:

$$|D_{actual} - D_{Desired}| < 2 * \left(\frac{10,000}{4,000,000} \right) = 0.005 \text{ (5,000 ppm)}$$

Additionally if the control loop rate does not evenly divide the switching period, the relative phase of the carrier will vary across switching cycles, and this error in the duty cycle will only be instantaneous, while the average error over neighboring switching cycles will likely be lower.

Given this information, in order to simplify the FPGA implementation of the sawtooth carrier, the carrier frequency is only changed once overflow is detected; in other words, the carrier frequency is not changed exactly at the beginning and end of the period of the sawtooth. As we have seen, the effect of the relative phase will be within known tolerances. Also, since the expected distribution range (the range of possible carrier frequencies) and control time step are relatively small, at worst it will slightly shift the average frequency (biased towards higher frequencies, since their overflow remainder will occupy proportionally more of the following cycle). If this is not desirable, the FPGA code could be modified to handle the overflow by incrementing the carrier proportionally based on the previous frequency, the amount of overflow, and the next frequency.

In order to evaluate the effect of the numerical representations of our reference and carrier signals on our output control signals, 16, 32, and 64-bit representations were considered. As seen in Figure 18, 16-bit resolution is insufficient, while 64-bit is even more wasteful than 32-bit. (Figure 14 illustrates the same plot when implemented with a 32-bit representation)

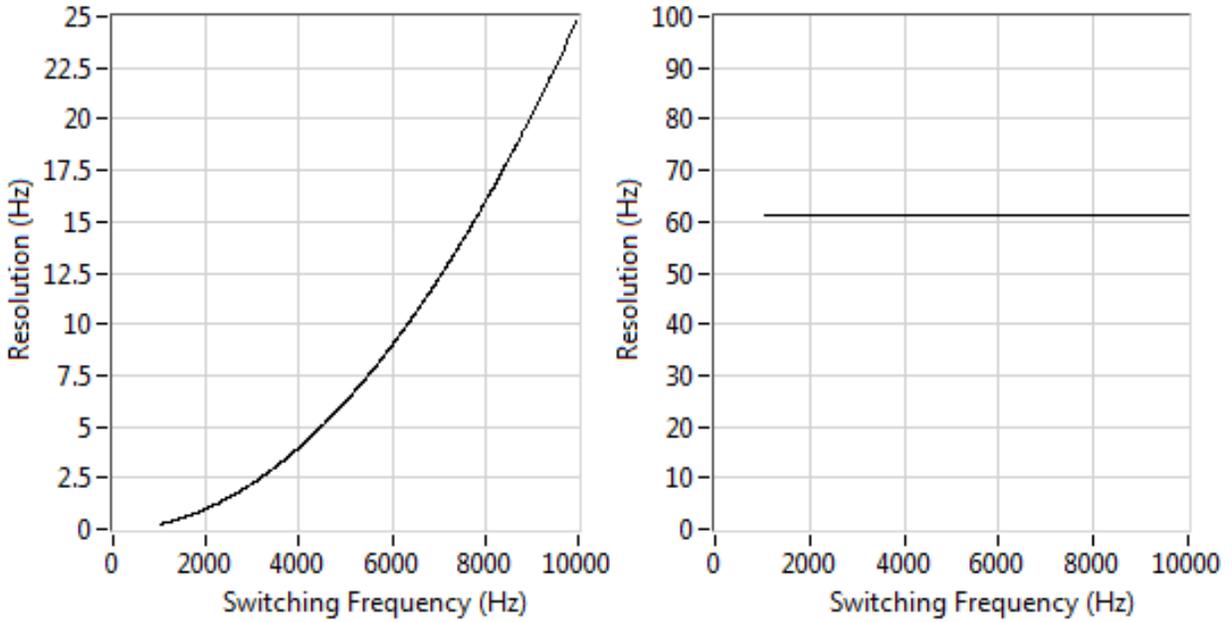


Figure 18. 64-bit, 250ns (left) , 16-bit, 250ns (right). The resolution of the 16-bit carrier frequency is insufficient; the frequency resolution is constant from 1000 to 10000Hz since the carrier resolution is determining the overall frequency resolution. The 64-bit representation offers no improvement over the 32-bit representation since both resolutions are more than sufficient, making frequency resolution completely dependent on control loop rate.

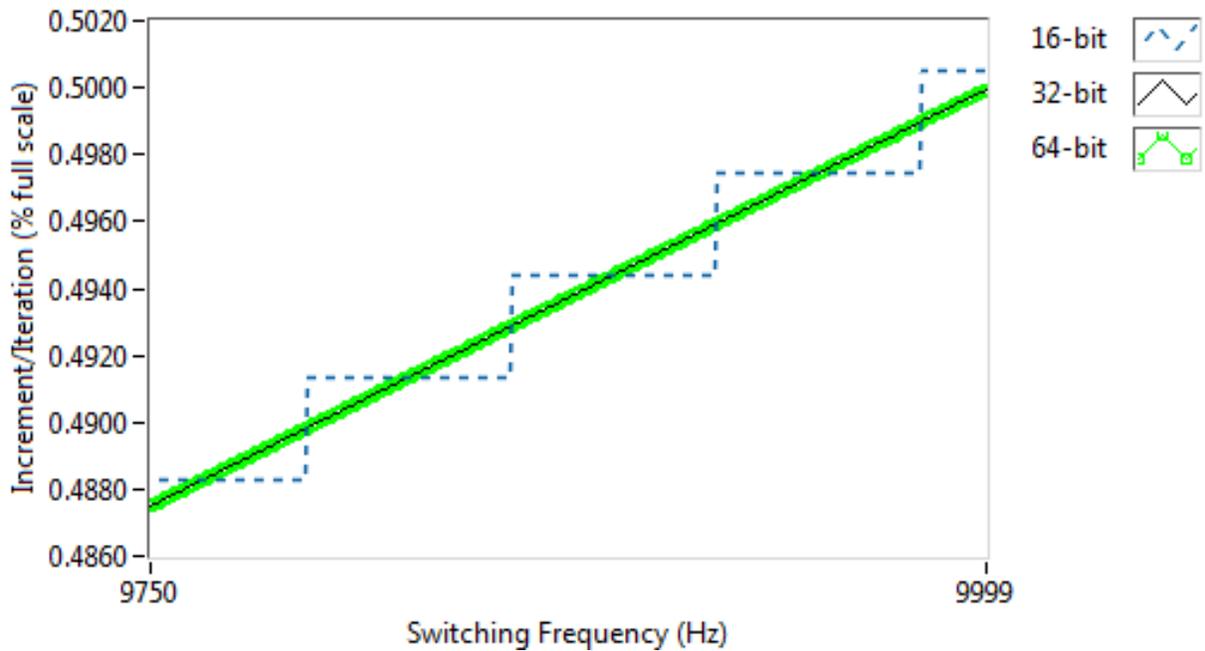


Figure 19. The 16-bit representation is does not provide sufficient resolution, while the 64-bit representation is not required since 32 bits provide all of the necessary resolution needed for our 4 MHz control loop rate.

As previously assumed, since the 32-bit representations provide more than sufficient resolution, the timing resolution is the primary decider of frequency resolution and duty cycle error. 16-bit representation is insufficient, 64-bit representation offers no improvement since 32-bit was already beyond our requirements. If one were so inclined, usage of fixed point numbers having resolution somewhere between 16 and 32 could also be investigated, but since we are not constrained by hardware in this case, we will just proceed with 32-bit representations.

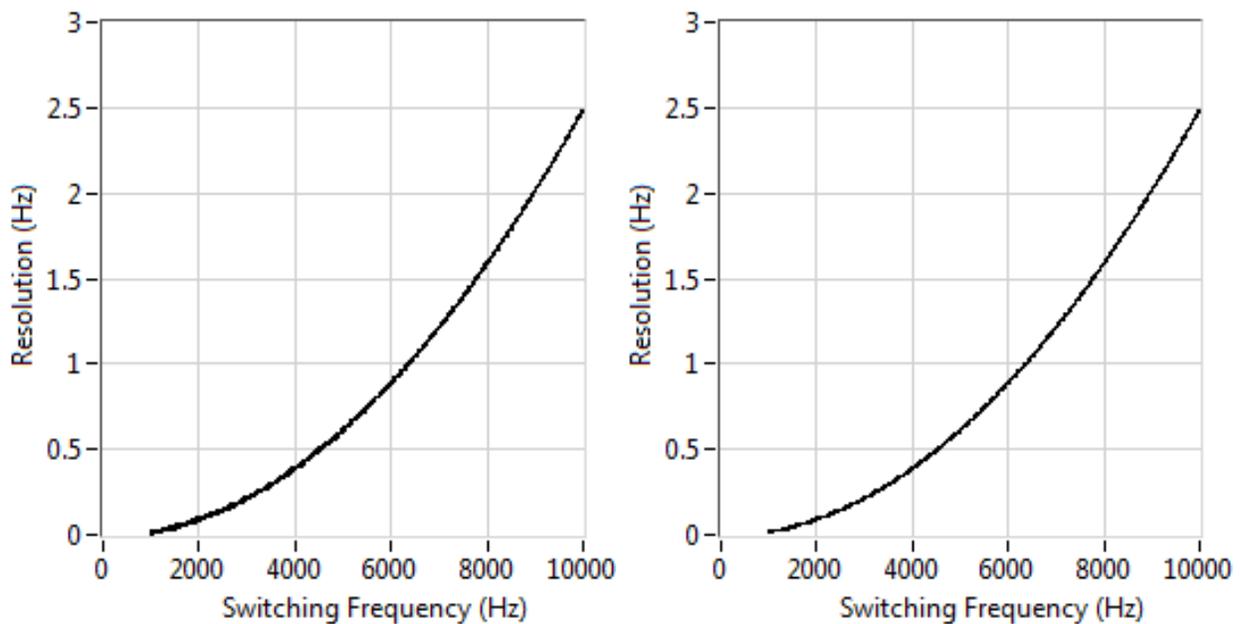


Figure 20. 25ns control loop period (40 MHz), 32-bit (left) and 64-bit (right). Even at 40 MHz, the loop rate seems to be the limiting factor on the frequency resolution (based on the fact that the shape of the curve conforms to the equation relating frequency resolution to control loop rate when not constrained by numerical representation resolution).

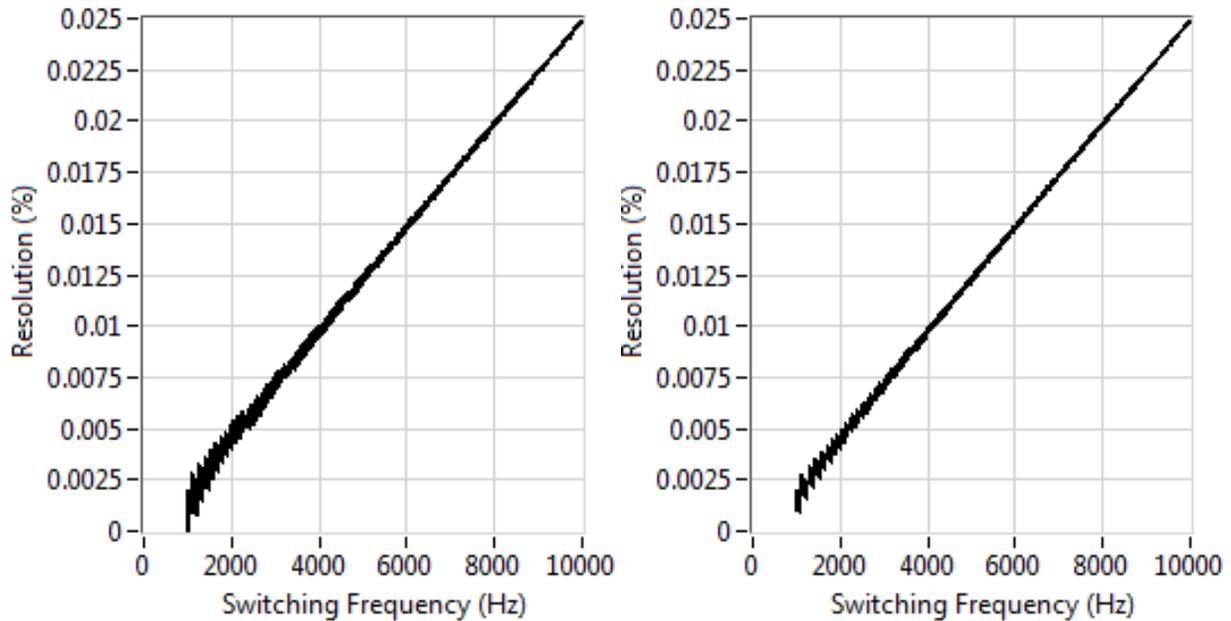


Figure 21. To better illustrate the effect of numerical representation and loop rate on the frequency resolution, the frequency resolution as a percentage of switching frequency is plotted versus switching frequency. 25ns control loop period, 32-bit (left) and 64-bit (right).

Figure 21 illustrates that at 40MHz control loop rate, the numerical resolution does slightly begin to affect frequency resolution, but only at very low switching frequencies, and only to a negligible extent. In order for the 32-bit representation to become the limiting factor, the control loop rate would need to be in excess of 40 MHz. For most sub-20 kHz motor drive applications, 32-bit representations will be more than sufficient.

To summarize, we could further increase the control loop rate in order to achieve better frequency and duty cycle fidelity, however by pipelining the code further, we introduce more latency. This latency may become unacceptable if the algorithm is ever modified for closed-loop control. In addition, we are going to be concentrating our attention to the 1 kHz to 6 kHz range for this study, so the 4 MHz loop rate will be sufficient.

Hardware overview

Hardware was selected in order to provide a robust control strategy prototyping system. While appendix A gives a detailed description of this system, Figure 22 shows all of the main components.

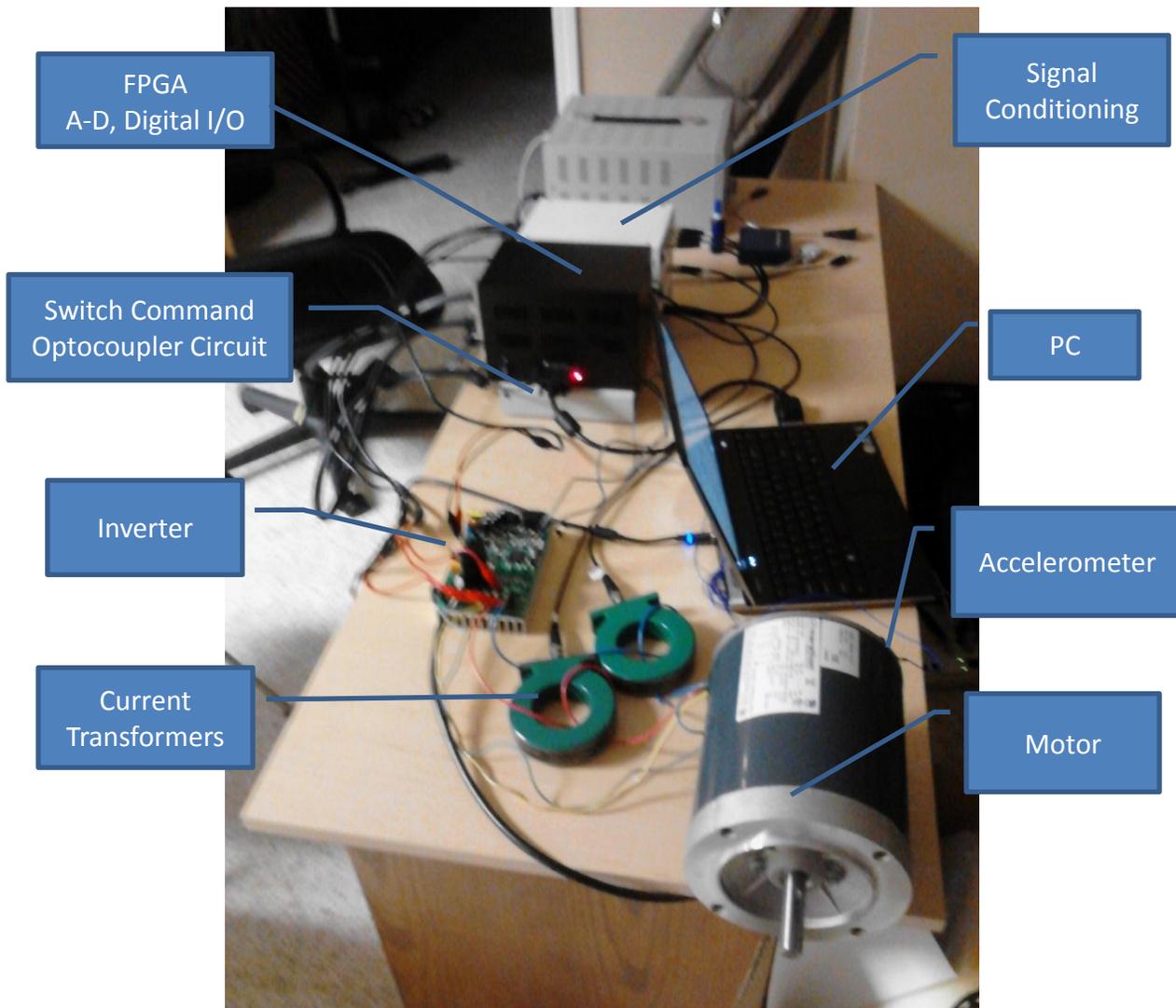


Figure 22. Picture of the entire system

Modeling and Simulation

In order to test the control algorithms, models of a 3-phase synchronous motor and inverter were employed. In order to perform the initial parameterization of the APRWM algorithm, a simple model of mechanical system response was also developed. All models and simulation programs were implemented using National Instruments LabVIEWtm [28].

Inverter Model

The inverter used in simulation and experiments for this study is a 3-phase two level voltage source inverter (Figure 23). Each leg of the inverter is modeled as indicated in Figure 24 and Equation 31. In order to accommodate deadtime simulation, the inverter model includes ideal diodes. During deadtime, one of the diodes will conduct, depending on the direction of the phase current. The complementary diode will not begin conducting once phase current reaches zero during deadtime.

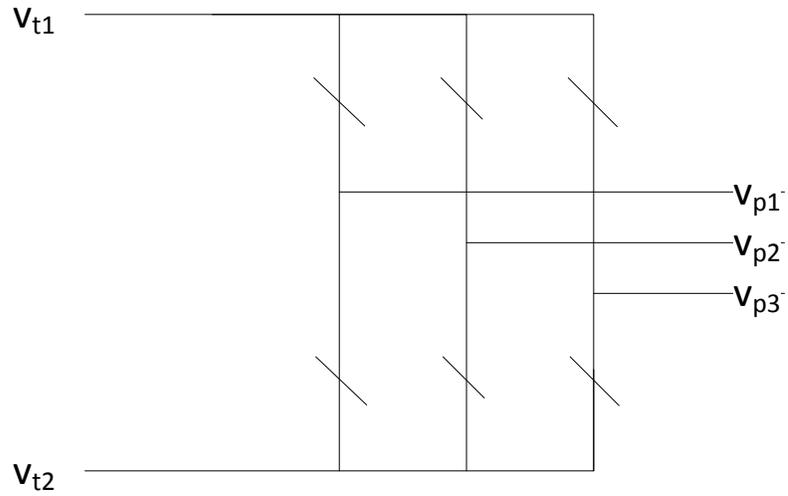


Figure 23. Topology of 3-phase inverter used in this study.

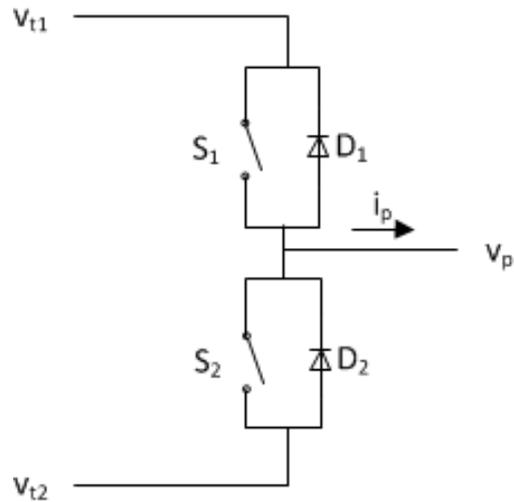


Figure 24. Inverter Model, Ideal Switches and Diodes

$$v_p = \begin{cases} v_{t1} & \text{if } S_1 \text{ or (deadtime and } i_p < 0, \text{ until } i_p \text{ reaches 0)} \\ v_{t2} & \text{if } S_2 \text{ or (deadtime and } i_p > 0, \text{ until } i_p \text{ reaches 0)} \\ v_p & \text{if deadtime and } i_p = 0 \text{ (floating terminal)} \end{cases} \quad (31)$$

Deadtime Model

In order to determine the interaction between deadtime and RPWM, deadtime coercion of the switch commands was employed during simulation. Figure 25 illustrates the effects of deadtime. The distortion resulting from deadtime is more significant at low speeds when the fundamental MI/phase voltage is lower. In cases where the modulation index is near 1, it is not as impactful.

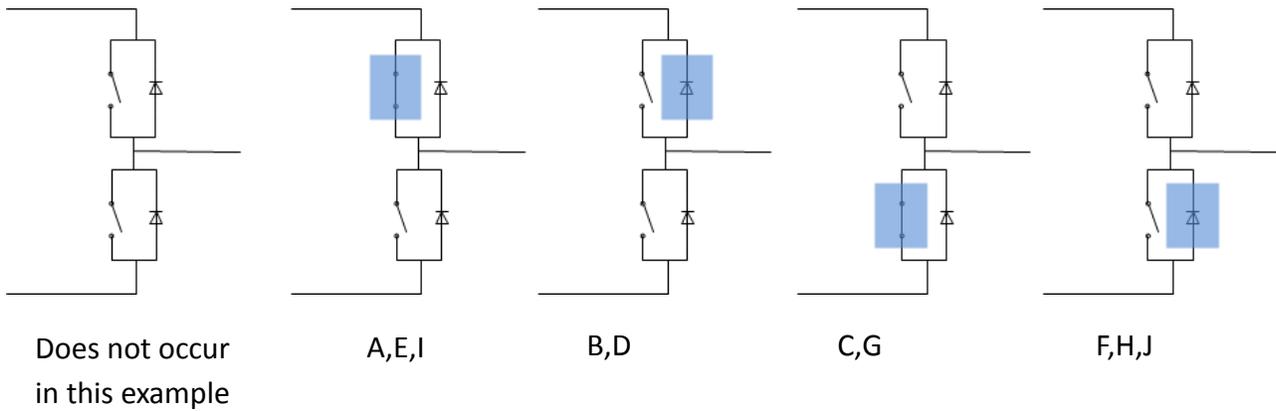
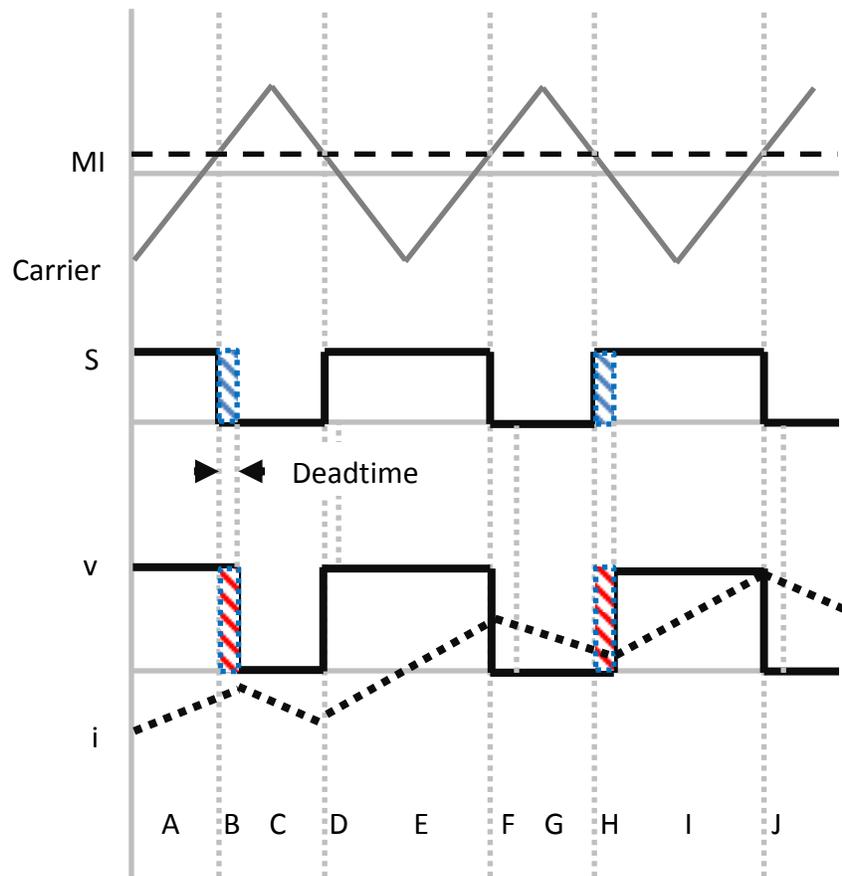


Figure 25. Illustration of inverter behavior during transitions in which deadtime influences the actual output voltage and introduces distortion. Each step (labeled A through J) is mapped to its corresponding state, indicating which component (if any) is conducting.

In a real system, non-ideal switch and diode behavior and current amplitude will cause slight variations in the actual behavior [29] [30] [31] [32], but this model should be sufficient for our purposes.

Most deadtime compensation approaches (such as [29] [30] [31]) employ some variation of modification of reference signal and/or switch transition times in order to realized the desired average volt-second output. This modification can occur within a single switching cycle, however it is sometimes necessary to compensate for volt-second error in neighboring switching cycles if compensation is not possible locally (due to the state of the system, compensation is not always possible near phase current zero crossings). Some of these techniques also include injection of additional content to the reference signal in order to cancel the distortion. Some of these approaches also utilize models of switching components to account for non-ideal switching. Most use a current measurement or at least a current polarity measurement in order to determine the correct compensation to be made.

In the case of RPWM, the similar transition offset techniques can be employed for deadtime compensation; the only difference is that some of the timing calculations will also be dependent on the instantaneous carrier frequency. But in general, the volt-second error can be tracked in the same manner and compensation will still involve adjustment of transition times. In the case of compensation through reference injection, additional considerations and analysis would need to be performed.

Motor Model

The 3-phase synchronous motor model (Figure 26) based on [33] assumes symmetric star windings, a non-salient rotor, and no misalignments in the magnetic circuit. The state space model (Equations 32,33,34,35) was implemented along with calculation of instantaneous flux linkage (sinusoidal function of rotor position) and neutral voltage at each time step. Along with neutral voltage calculation, provisions were included to allow terminal voltages to float.

$$A = \begin{bmatrix} -\frac{R}{L} & 0 & 0 & -\frac{\lambda * f_a(\theta)}{L} & 0 \\ 0 & -\frac{R}{L} & 0 & -\frac{\lambda * f_b(\theta)}{L} & 0 \\ 0 & 0 & -\frac{R}{L} & -\frac{\lambda * f_c(\theta)}{L} & 0 \\ \frac{\lambda * f_a(\theta)}{J} & \frac{\lambda * f_b(\theta)}{J} & \frac{\lambda * f_c(\theta)}{J} & -\frac{B}{J} & 0 \\ 0 & 0 & 0 & \frac{P}{2} & 0 \end{bmatrix} \quad (32)$$

$$B = \begin{bmatrix} \frac{1}{L} & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & 0 \\ 0 & 0 & \frac{1}{L} & 0 \\ 0 & 0 & 0 & -\frac{1}{J} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (33)$$

$$x(t) = \begin{bmatrix} i_a \\ i_b \\ i_c \\ \omega \\ \theta \end{bmatrix} \quad (34)$$

$$u(t) = \begin{bmatrix} v_a \\ v_b \\ v_c \\ T_m \end{bmatrix} \quad (35)$$

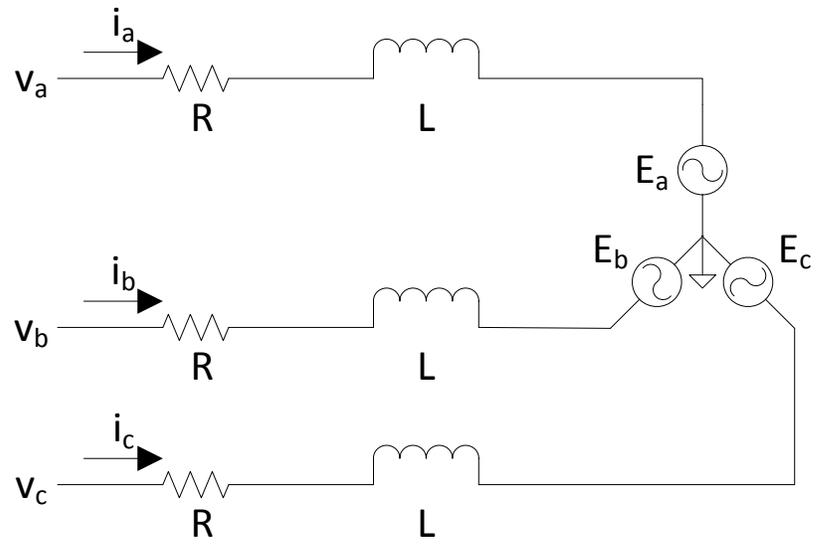


Figure 26. 3-phase synchronous motor model equivalent circuit

Simulation Results

Simulations were performed in order to compare the spectra of the phase currents using SPWM and various RPMW techniques, to compare spectra in the presence of deadtime, and to perform preliminary evaluations on the ARPWM gradient descent algorithm. Figure 27 shows an example of the time histories resulting from a simulation.

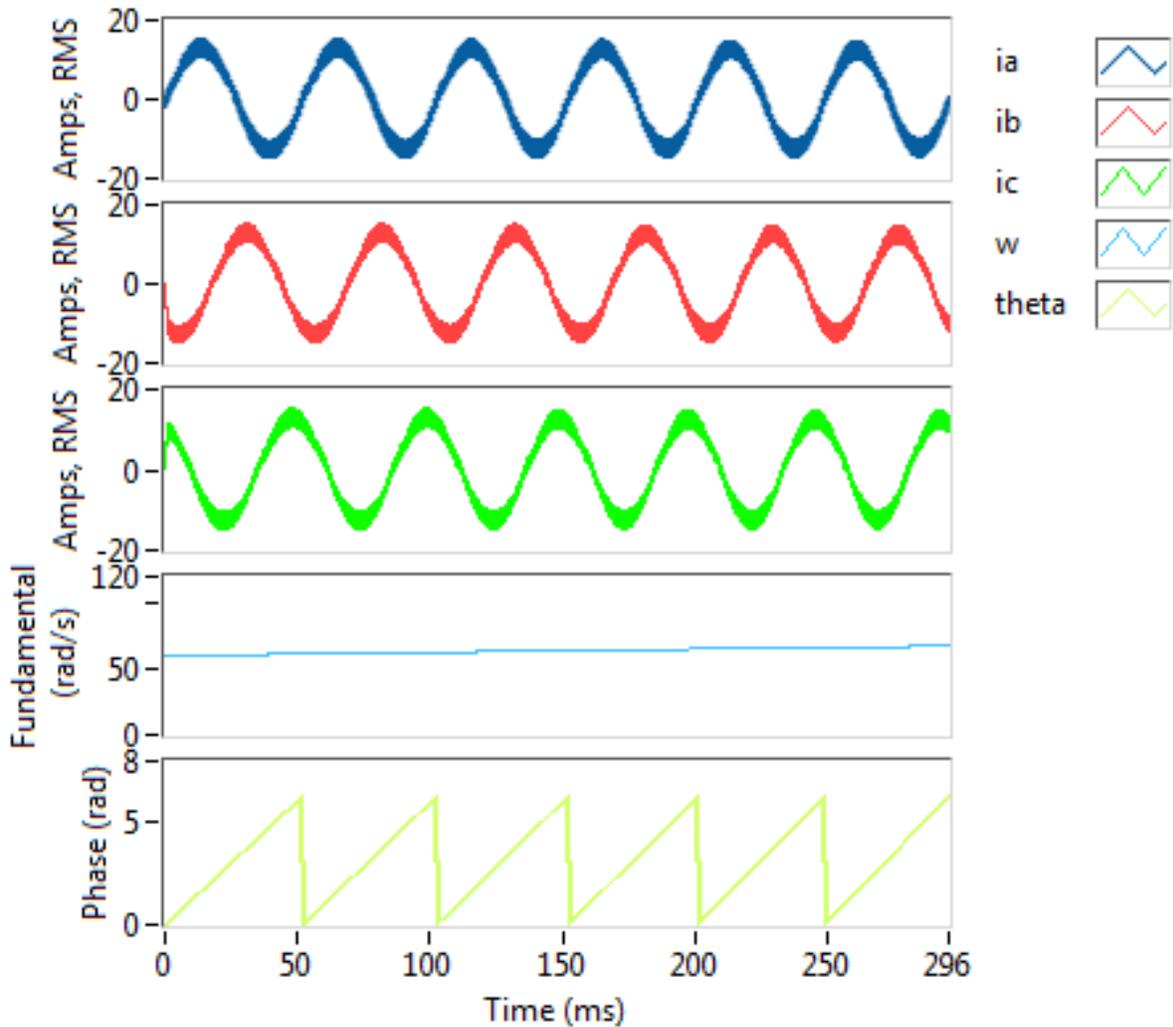


Figure 27. Example simulation time histories.

Simulation Comparison of SPWM and various RPWM techniques

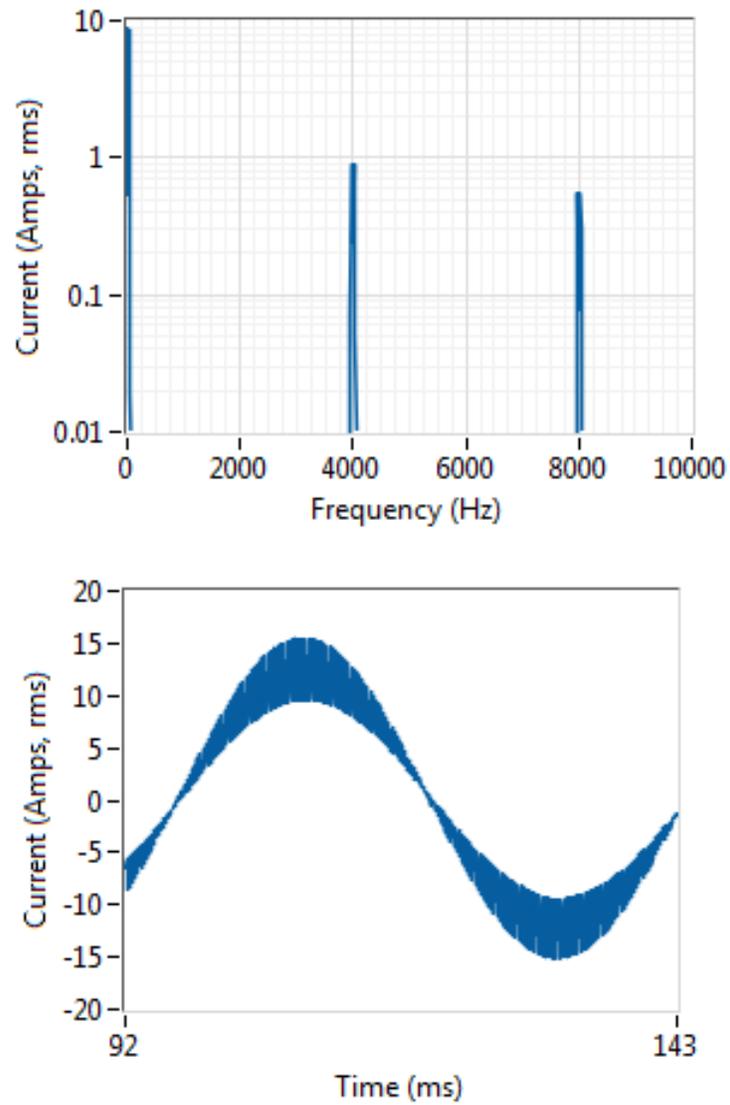


Figure 28. Frequency-domain phase current (left) and time-domain phase current (right): SPWM @ 4 kHz

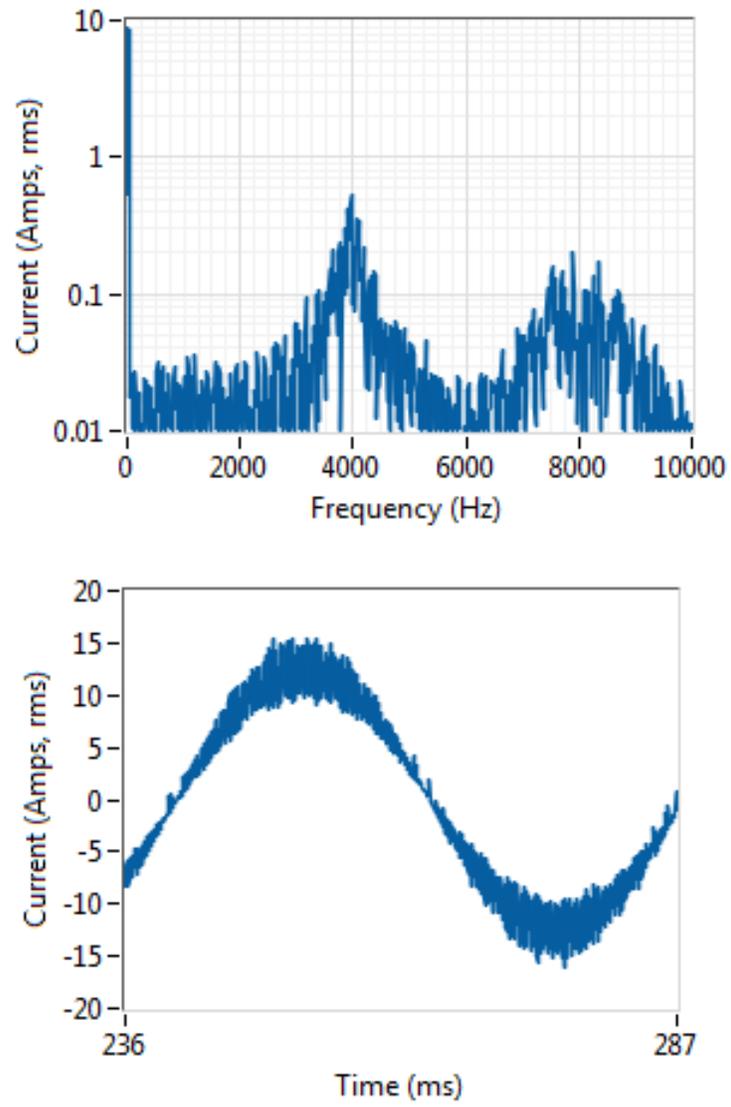


Figure 29. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random carrier frequency, 4 kHz center frequency, 1 kHz range.

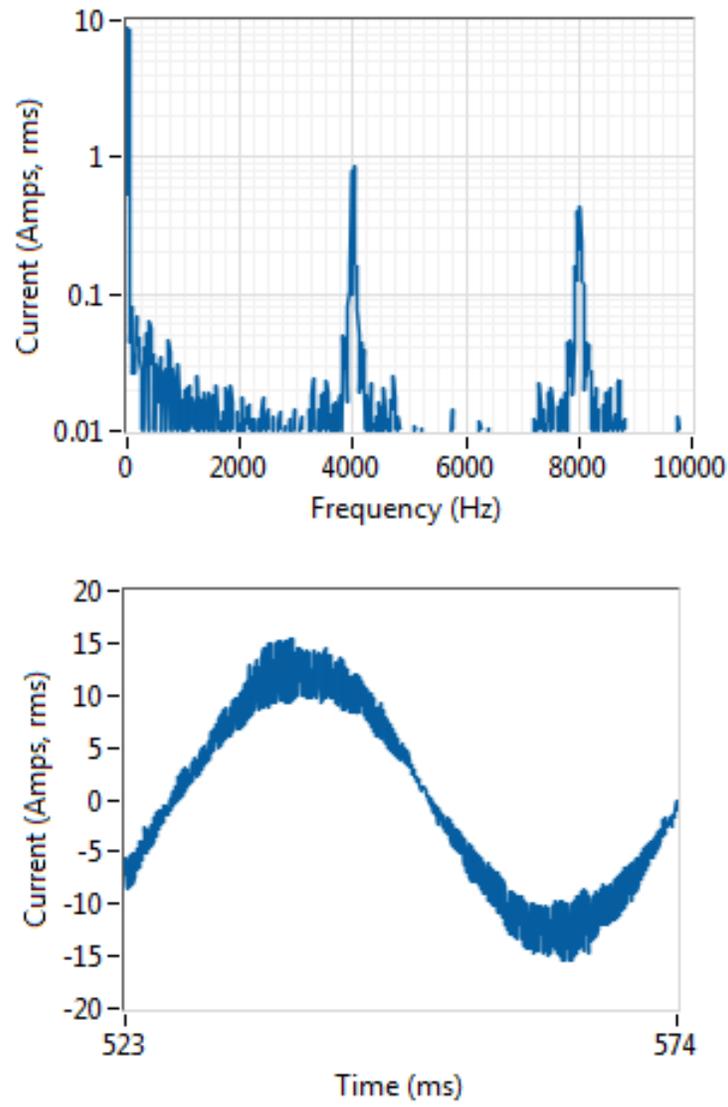


Figure 30. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random carrier function, 4 kHz center frequency, 10 kHz instantaneous range. The range is larger for the random function because the average range over a switching cycle is much less than the instantaneous range.

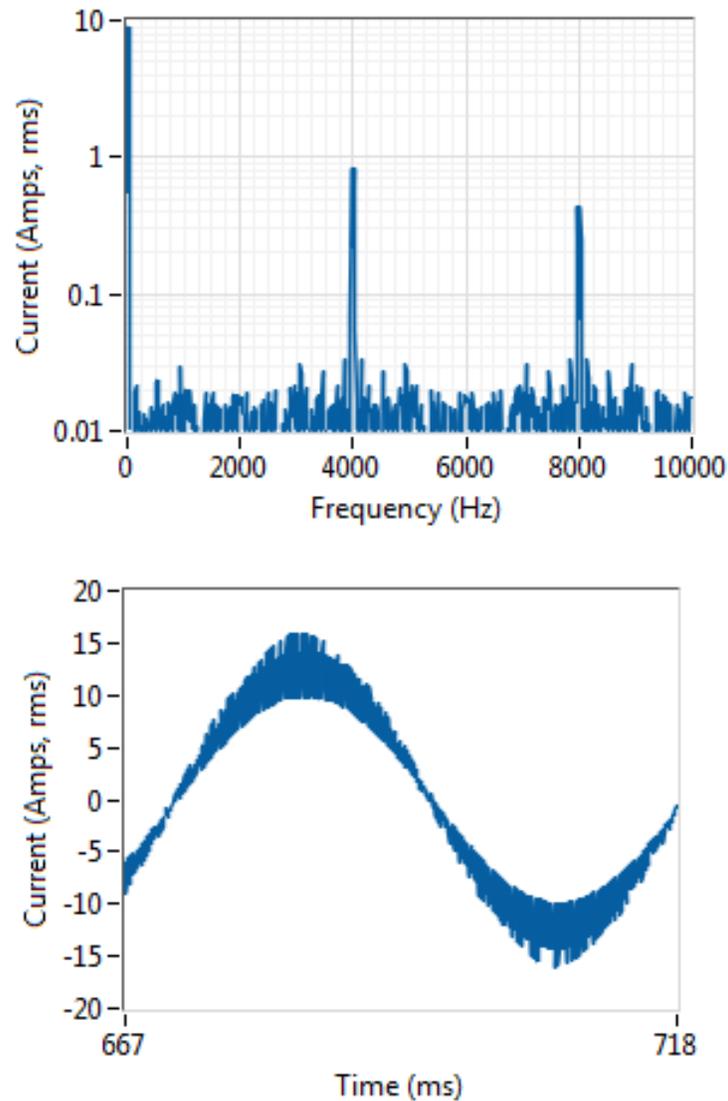


Figure 31. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random pulse position, 4 kHz center frequency, random selection of leading or trailing on-time transition.

Of each of the techniques, the pulse position results in the highest current ripple. If we were to improve the random pulse position by allowing the pulse to begin at any time step in the switching cycle, rather than at the beginning or end, there would be improvement in the typical ripple amplitude. As expected, the peak 1st and 2nd order ripple are reduced as the range of

the random distribution is increased, regardless of the approach. RPWM with random carrier frequency is preferred for this study since it seems to provide the most consistent ripple behavior and is the easiest to implement in FPGA due to the relaxed requirement of only needing to calculate a new carrier frequency once per switching cycle.

Simulation of SPWM and RPWM behavior with Deadtime

Comparison graphics were generated to evaluate the interaction between deadtime and RPWM.

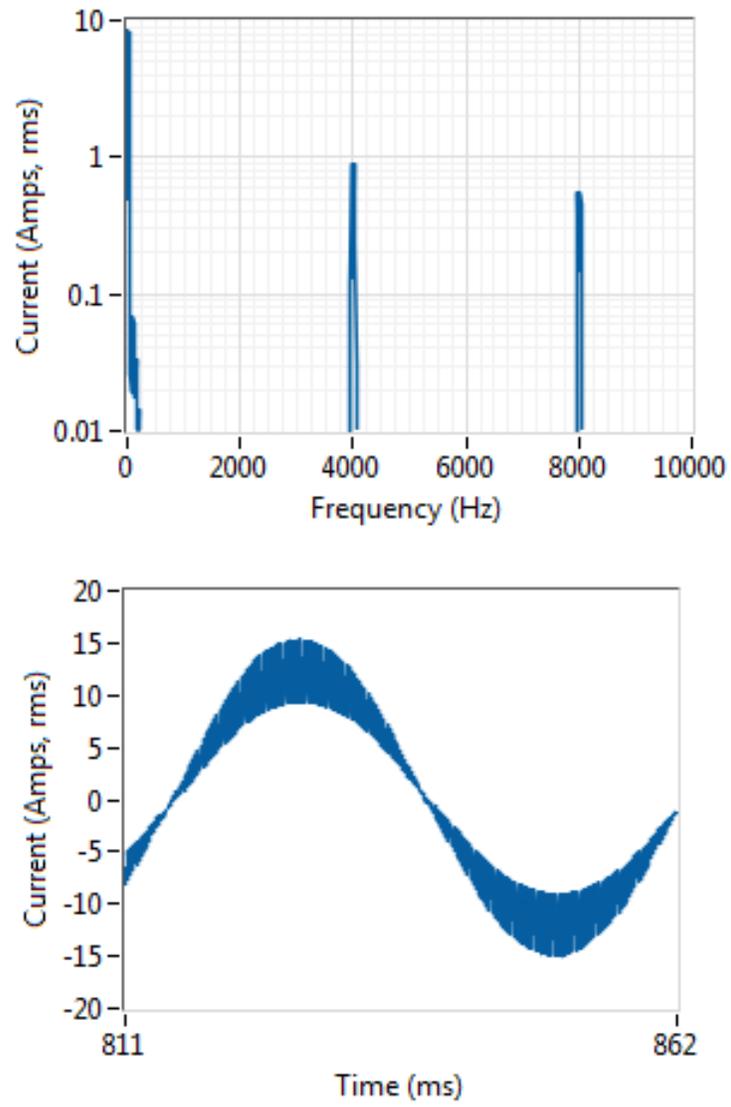
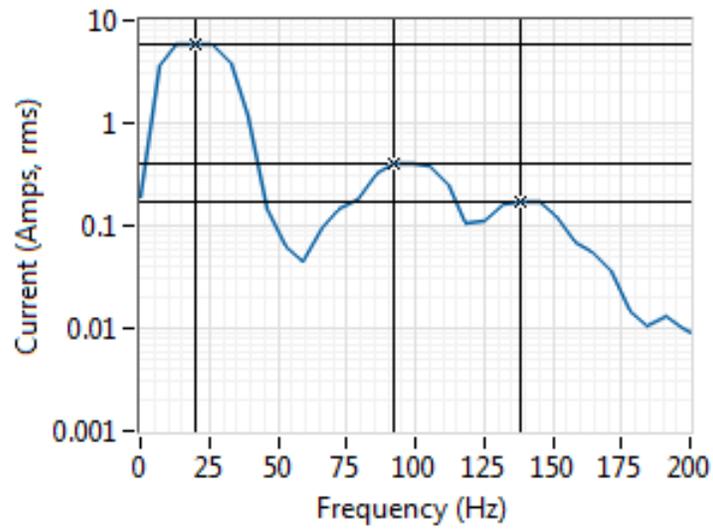


Figure 32. Frequency-domain phase current (left) and time-domain phase current (right): SPWM @ 4 kHz, 120 rad/s fundamental frequency, 500ns Deadtime



Cursors:		X	Y
<input type="checkbox"/>	Fundamental		
	Plot 0	19.7694	5.92092
<input type="checkbox"/>	5th		
	Plot 0	92.257	0.411121
<input type="checkbox"/>	7th		
	Plot 0	138.386	0.166406

Figure 33. Notice the extra content at multiples (5^{th} order at about 100 Hz and 7^{th} order at about 140 Hz) of the fundamental (~ 20 Hz) due to deadtime distortion with SPWM.

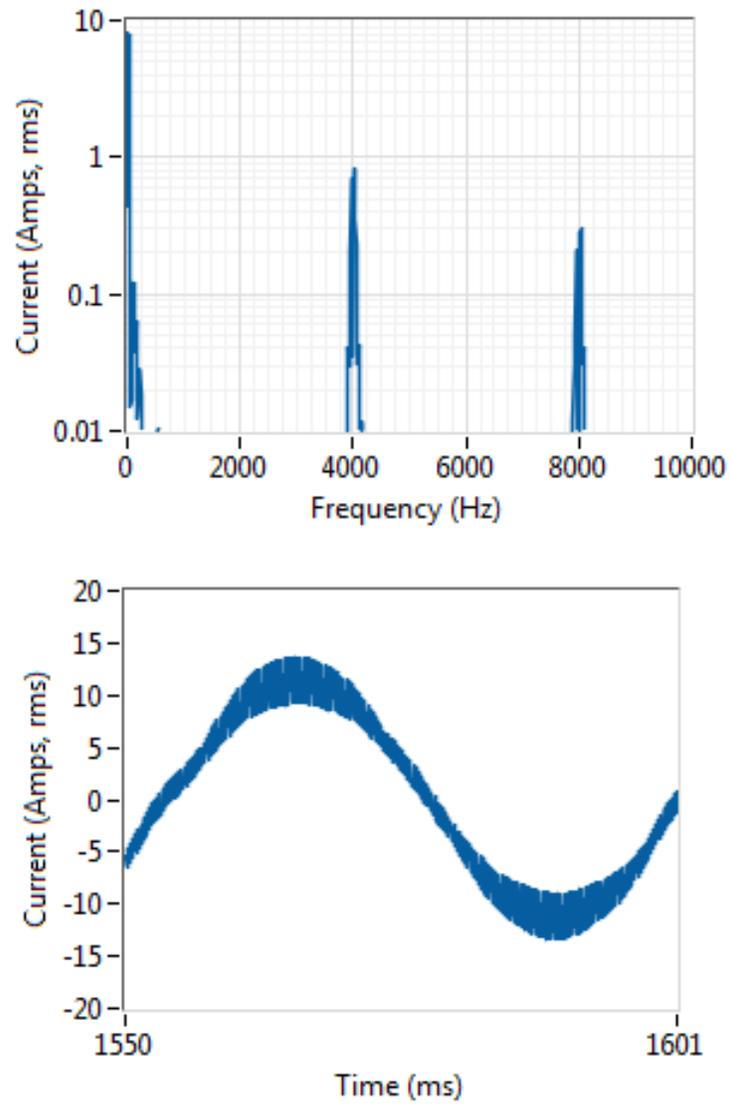


Figure 34. Frequency-domain phase current (left) and time-domain phase current (right): SPWM @ 4 kHz, 60 rad/s fundamental frequency, 1000ns Deadtime

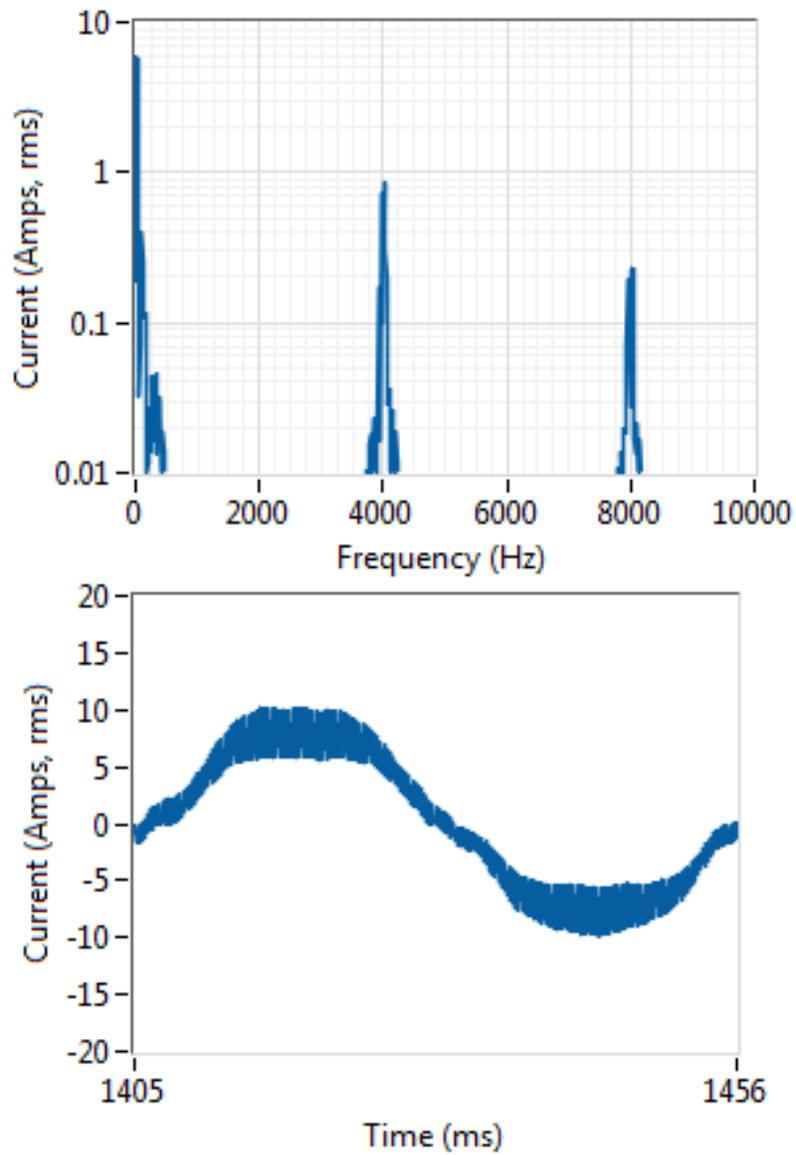


Figure 35. Frequency-domain phase current (left) and time-domain phase current (right): SPWM @ 4 kHz, 60 rad/s fundamental frequency, 4000ns Deadtime

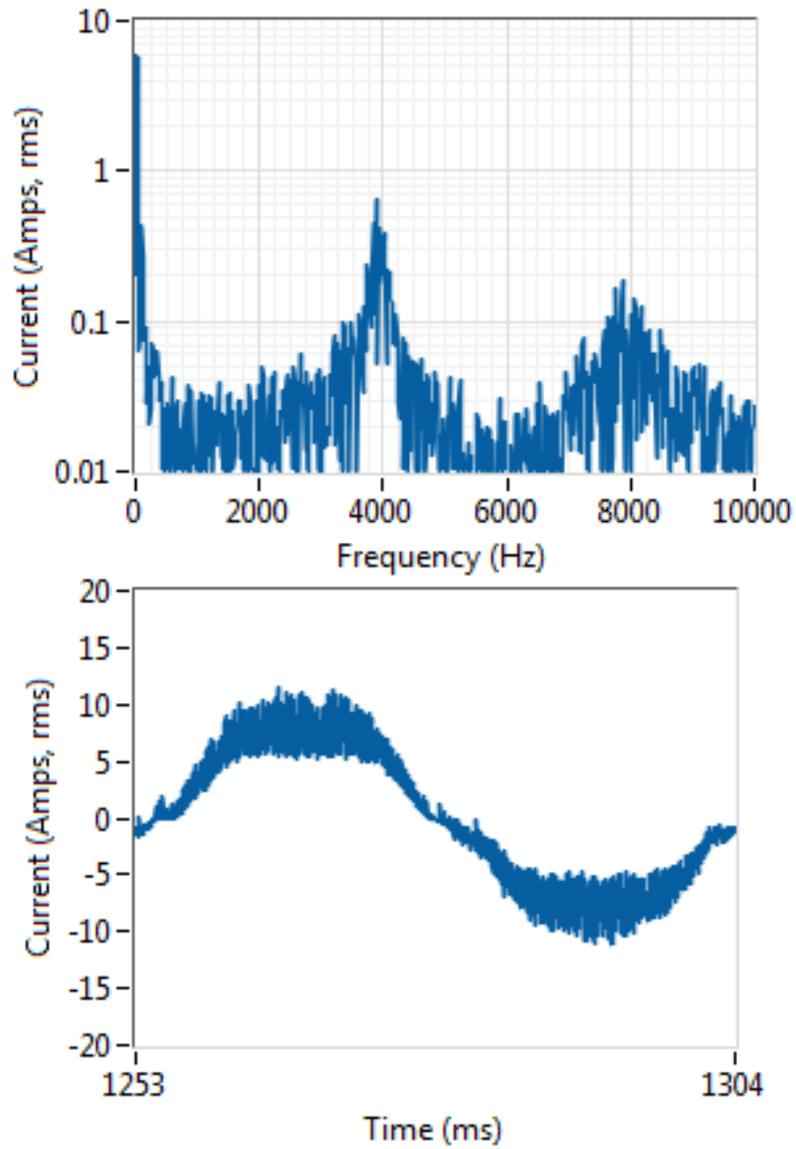


Figure 36. Frequency-domain phase current (left) and time-domain phase current (right):
 RPWM with random carrier frequency, 4 kHz center frequency, 1 kHz range, 60 rad/s
 fundamental frequency, 4000ns Deadtime

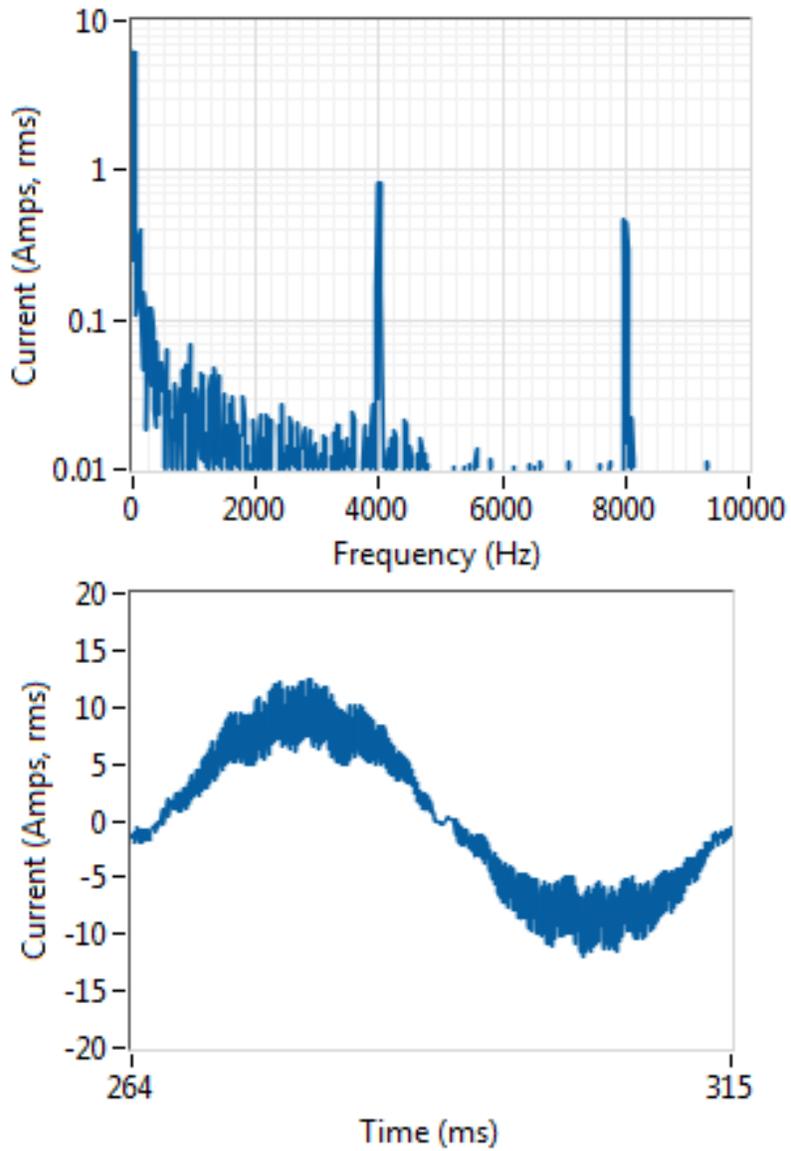
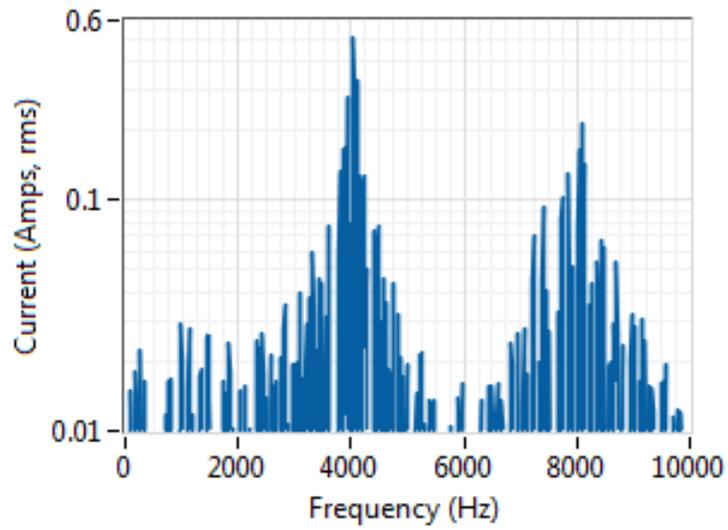
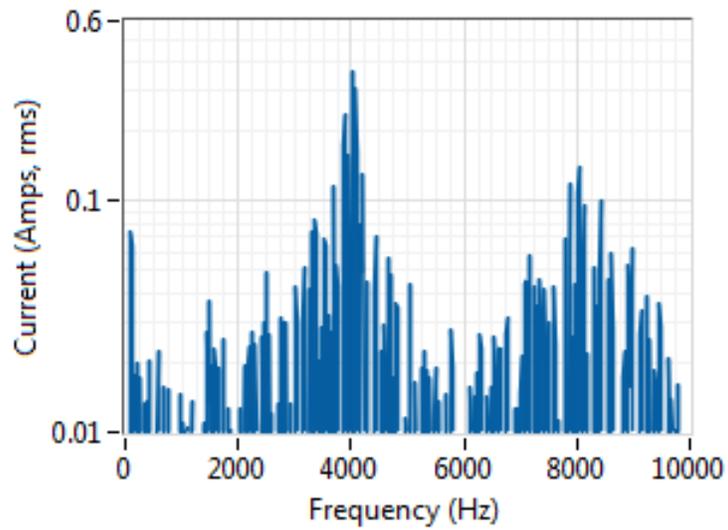


Figure 37. Frequency-domain phase current (left) and time-domain phase current (right): RPWM with random pulse position, 4 kHz center frequency, 60 rad/s fundamental frequency, 4000ns Deadtime. Notice the improvement in distortion.



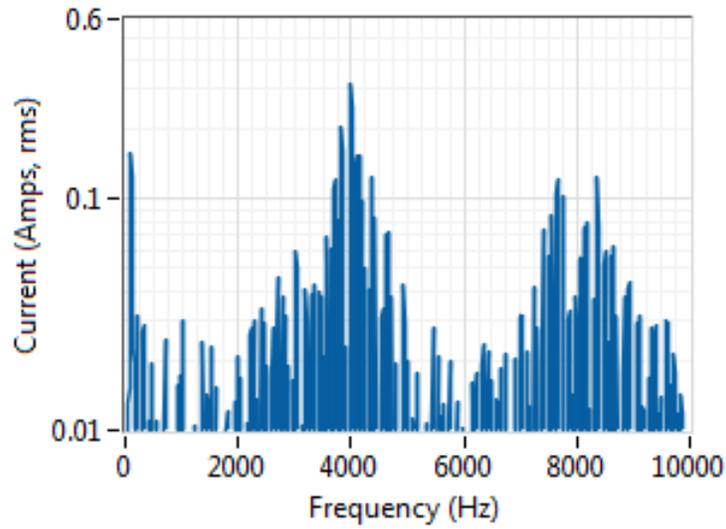
Another 0 deadtime run



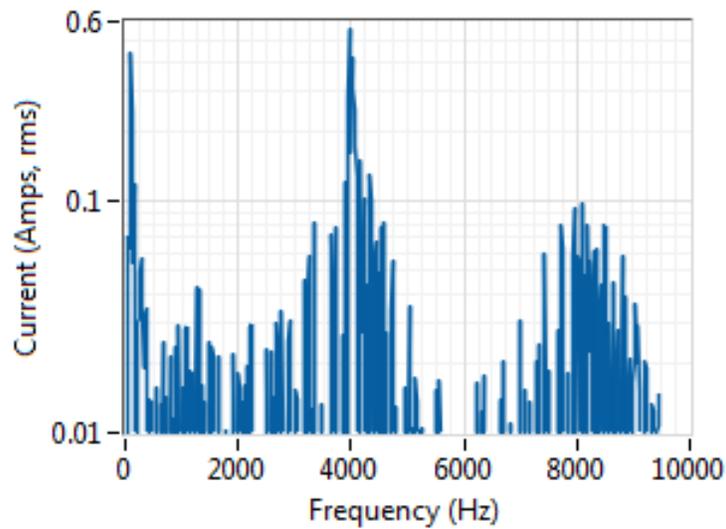
500ns deadtime

Figure 38. The Difference in phase current amplitude between simulations performed at various deadtime values and zero deadtime. RPWM with random carrier frequency, 4 kHz center frequency with 1 kHz range.

Figure 38 (cont'd)



1000ns deadtime



4000ns deadtime

In order to better illustrate the possible effect of increasing deadtime on the spectral content around switching frequency orders,

Figure 38 includes the frequency domain graphics for the difference in amplitudes from the case with no deadtime. It is clear that deadtime has a negligible, if any, effect on the content at

switching frequency orders. The variation of peak amplitude at orders of switching frequency due to the stochastic nature of RPWM is far greater than any variation introduced while increasing deadtime.

The deadtime does not significantly affect the phase current near switching frequencies; but there is reduction in deadtime distortion when we use random pulse position RPWM. This is because the random pulse position approach results in a sawtooth carrier at the intended switching frequency or the equivalent of a triangle carrier at half of the intended switching frequency – which is why the deadtime distortion is reduced (since there are fewer transitions and few opportunities for volt-second error) There also seems to be a slight increase in THD using either RPWM with random carrier frequency, or RPWM with random carrier function.

Simulation of ARPWM

In order to test and parameterize the gradient descent algorithm, and to test the objective function of gradient descent algorithm, a simple system mechanical response model was employed (Figure 39).

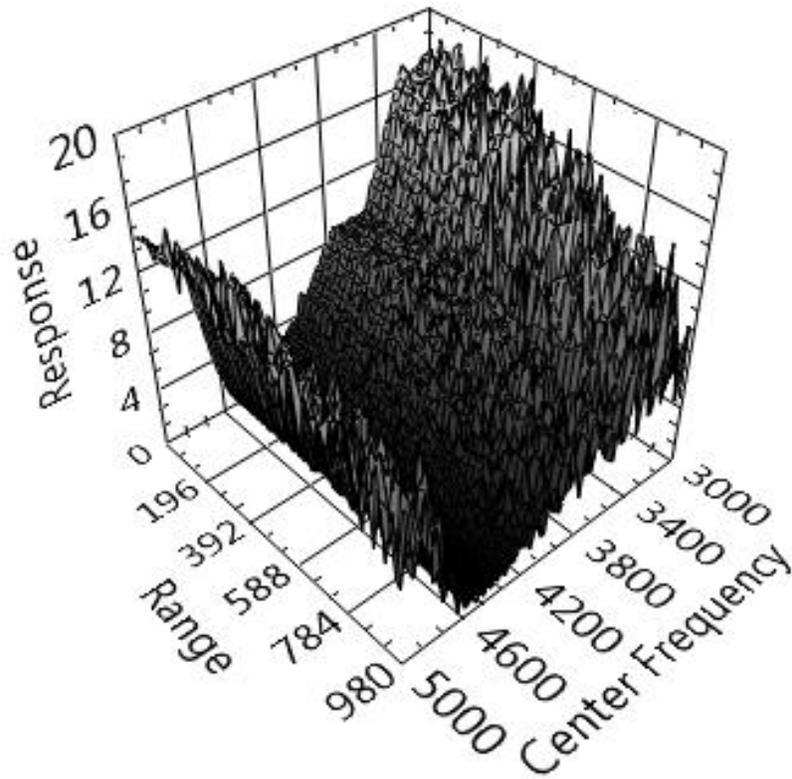
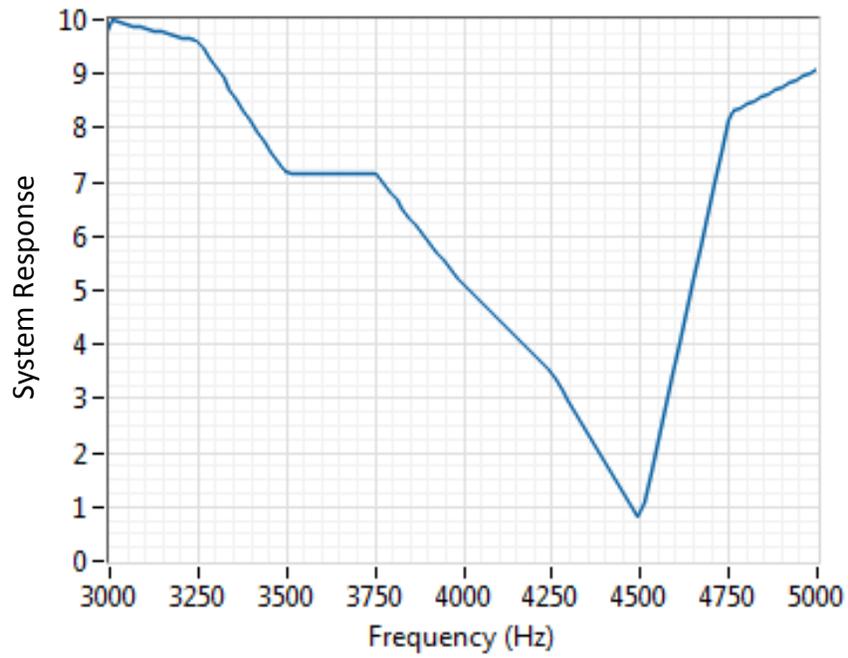


Figure 39. Example simulated system response (left) and corresponding search space (right). The search space is the value of our objective function at each center frequency/range combination over the frequency range of interest.

A few different scenarios were used for preliminary testing of the ARPWM gradient descent algorithm. In Figure 40, the modeled system response signature includes a smooth valley with moderate response levels in search space. Notice how it is advantageous to have a large range to spread the energy across the valley.

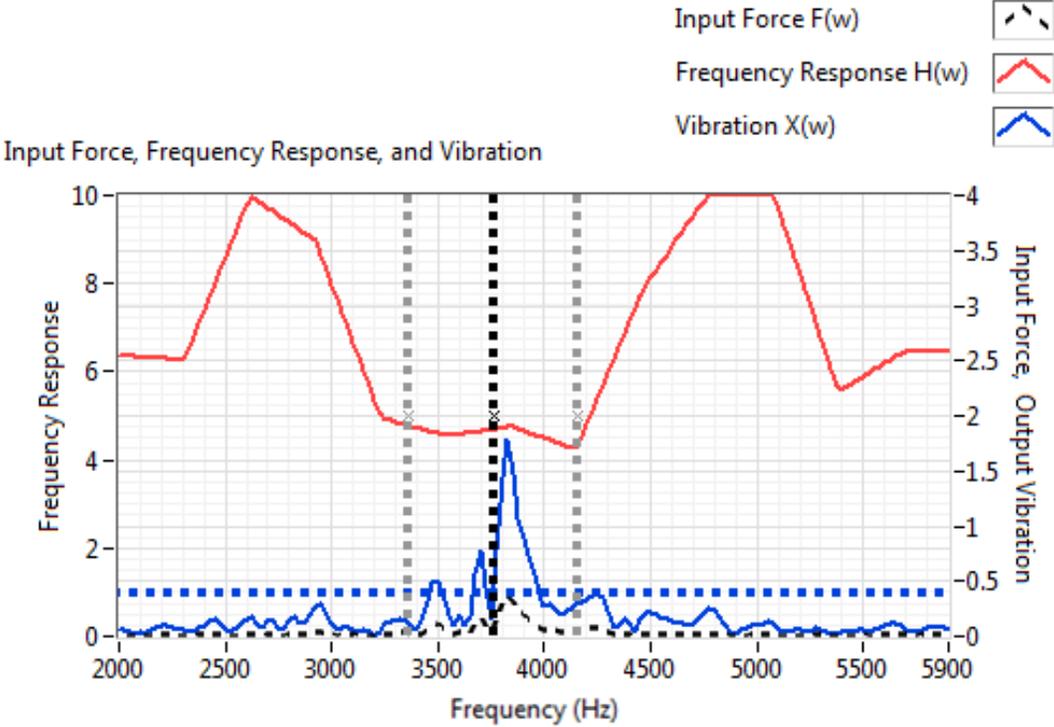


Figure 40. Simulated ARPWM final steady state distribution in presence of smooth valley with moderate response levels in search space.

Figure 41 depicts a simulated response containing a sharp valley with low response levels. Here, the range has tended towards zero in order to avoid exciting the neighboring high-gain regions. Figure 42 demonstrates the final steady state distribution after the sharp valley of Figure 41 is adjusted into flat valley with a center shifted above 4kHz. Now the response has changed and the center frequency and range have adjusted accordingly.

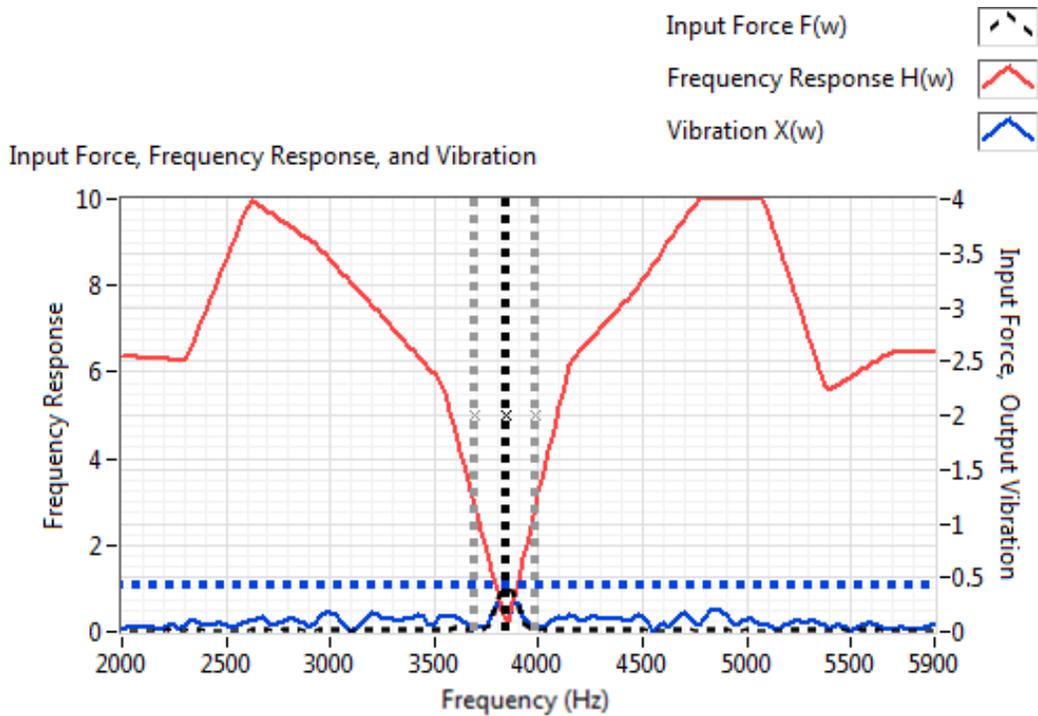


Figure 41. Simulated ARPWM final steady state distribution in presence of sharp valley with low response levels in search space.

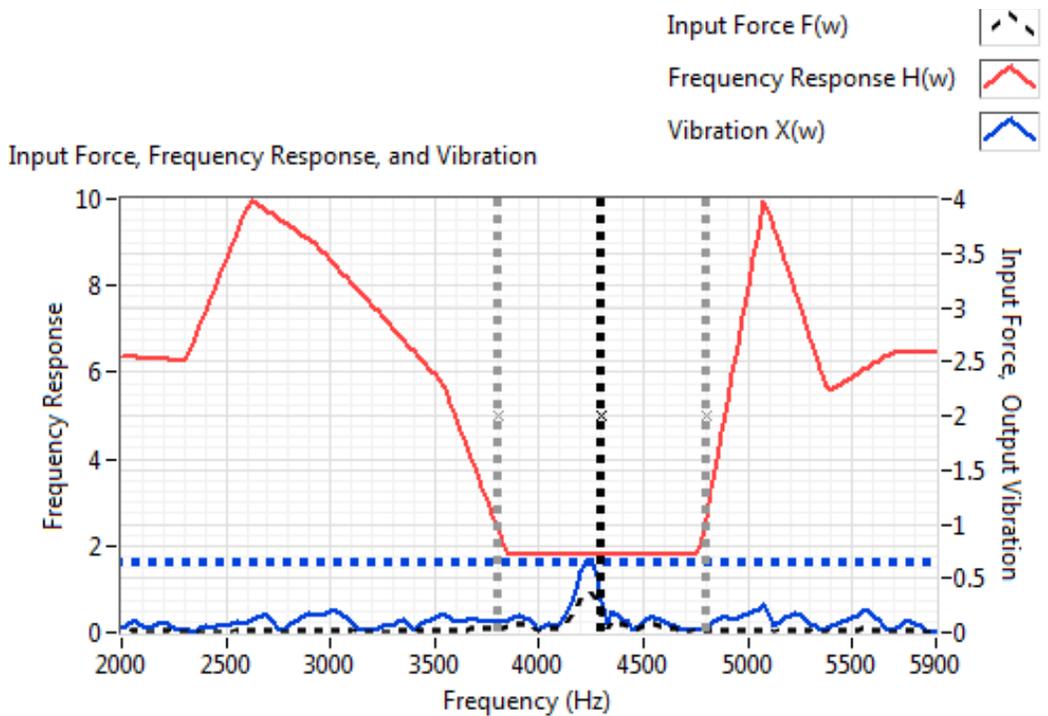


Figure 42. Simulated ARPWM final steady state distribution after sharp valley is adjusted to flat valley and center of valley is shifted above 4kHz.

Experiments and Results

Initially tests were performed in order to map out our search space – so we can judge the performance of the gradient descent algorithm based on the known characteristics of the search space.

A program (Figure 43) was written to iterate through several different switching frequency distributions (center frequency and range combinations). The gradient descent objective function value was averaged at each condition in order to be able to plot the search space in three dimensions.

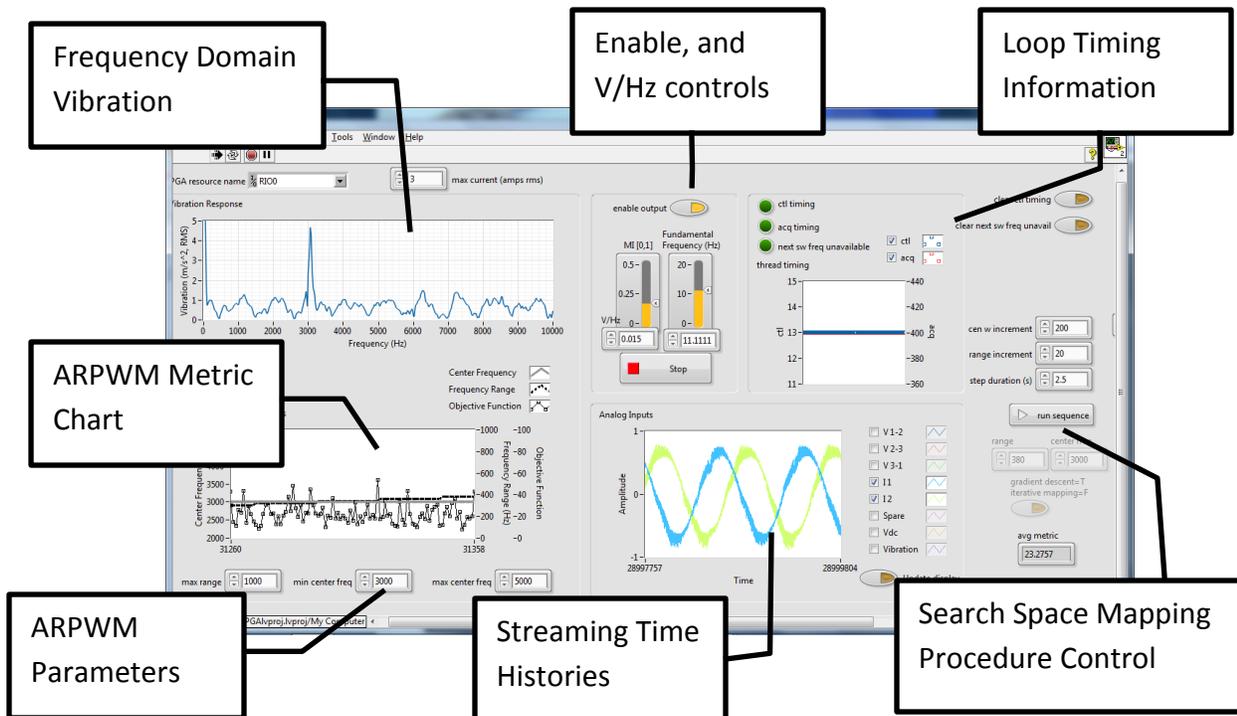


Figure 43. Screenshot of the program written to iterate through and map the search space.

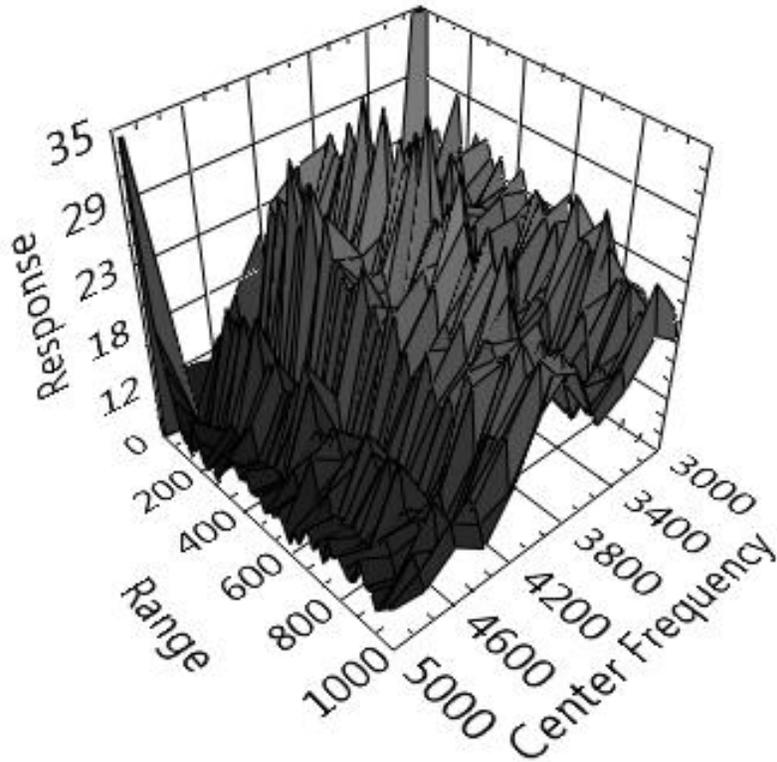


Figure 44. Search space at 10 Hz fundamental frequency, no load

Figure 44 depicts the search space presented to the ARPWM gradient descent algorithm. The surface represents the value of our objective function at each center frequency/range combination over the frequency range of interest. The search space from 3000 Hz to 5000 Hz center frequency (in 200 Hz increments) and 0 to 1000 Hz range (in 20Hz increments). The response is still decreasing after the range reaches 1000 Hz; the experiment will repeated up to 2 kHz range.

The time-histories of various ARPWM gradient descent parameters along with a time-frequency plot of the system response is included in Figure 45.

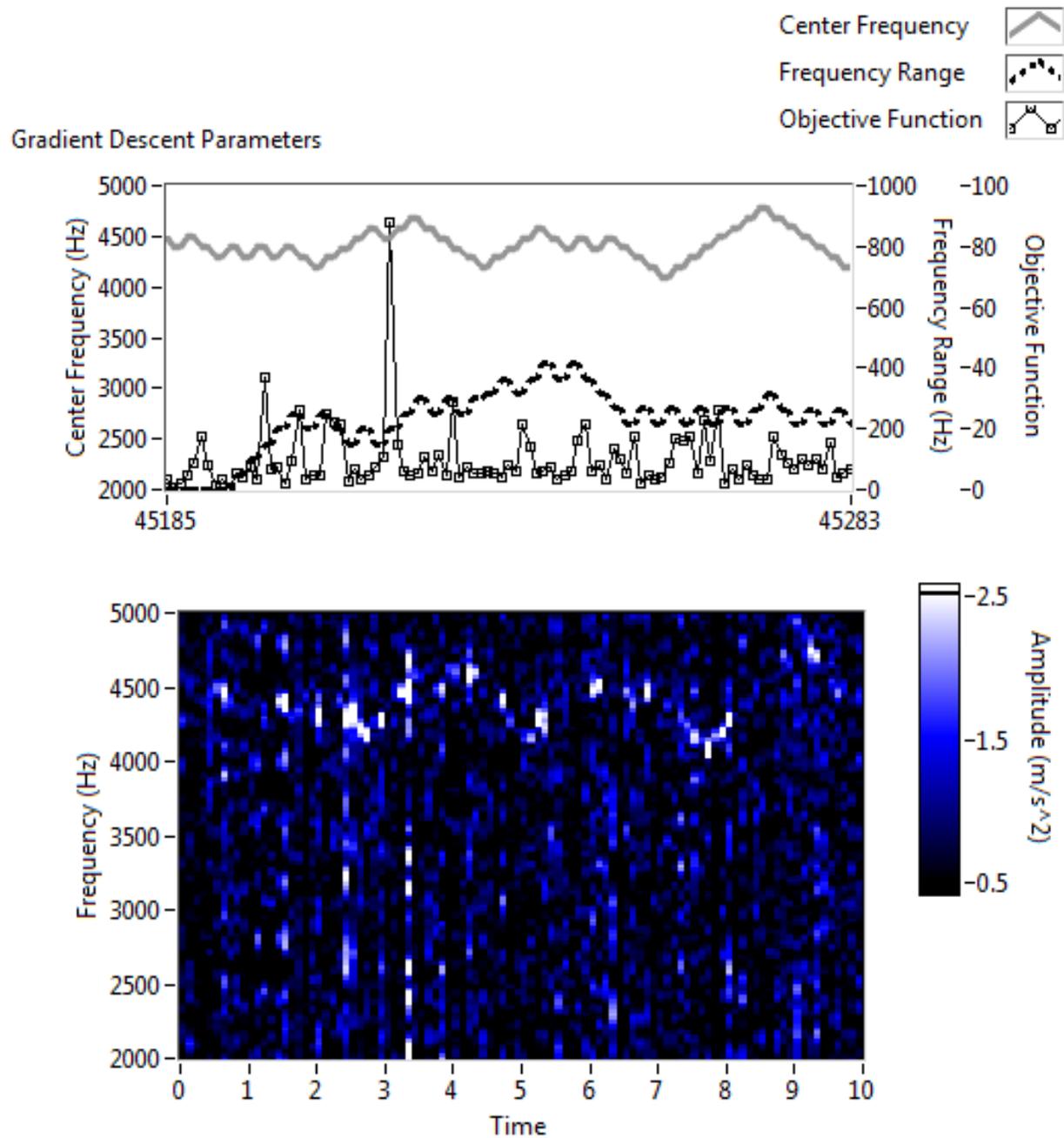


Figure 45. ARPWM over time. 11 Hz fundamental frequency.

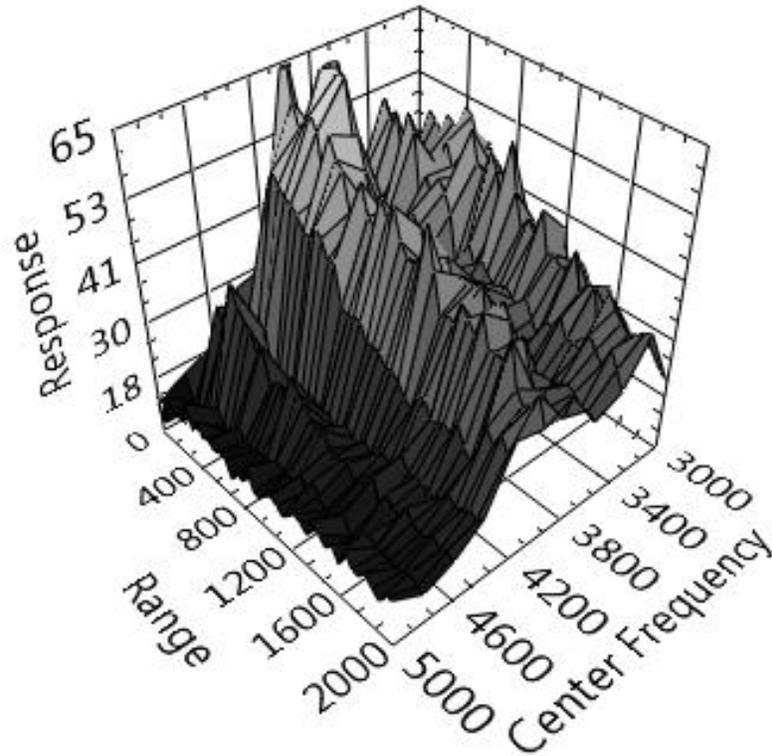


Figure 46. Search space, 20 Hz fundamental, no load.

Figure 46 is similar to Figure 44 however the fundamental frequency of the reference signal has been increased to 20 Hz in order to demonstrate that there has been a change in the system mechanical response. Additionally, our iterative mapping of the search space has been expanded to include distribution ranges up to 2 kHz.

Figure 47 presents the time-histories of various ARPWM gradient descent parameters along with a time-frequency plot of the system response under the new conditions.

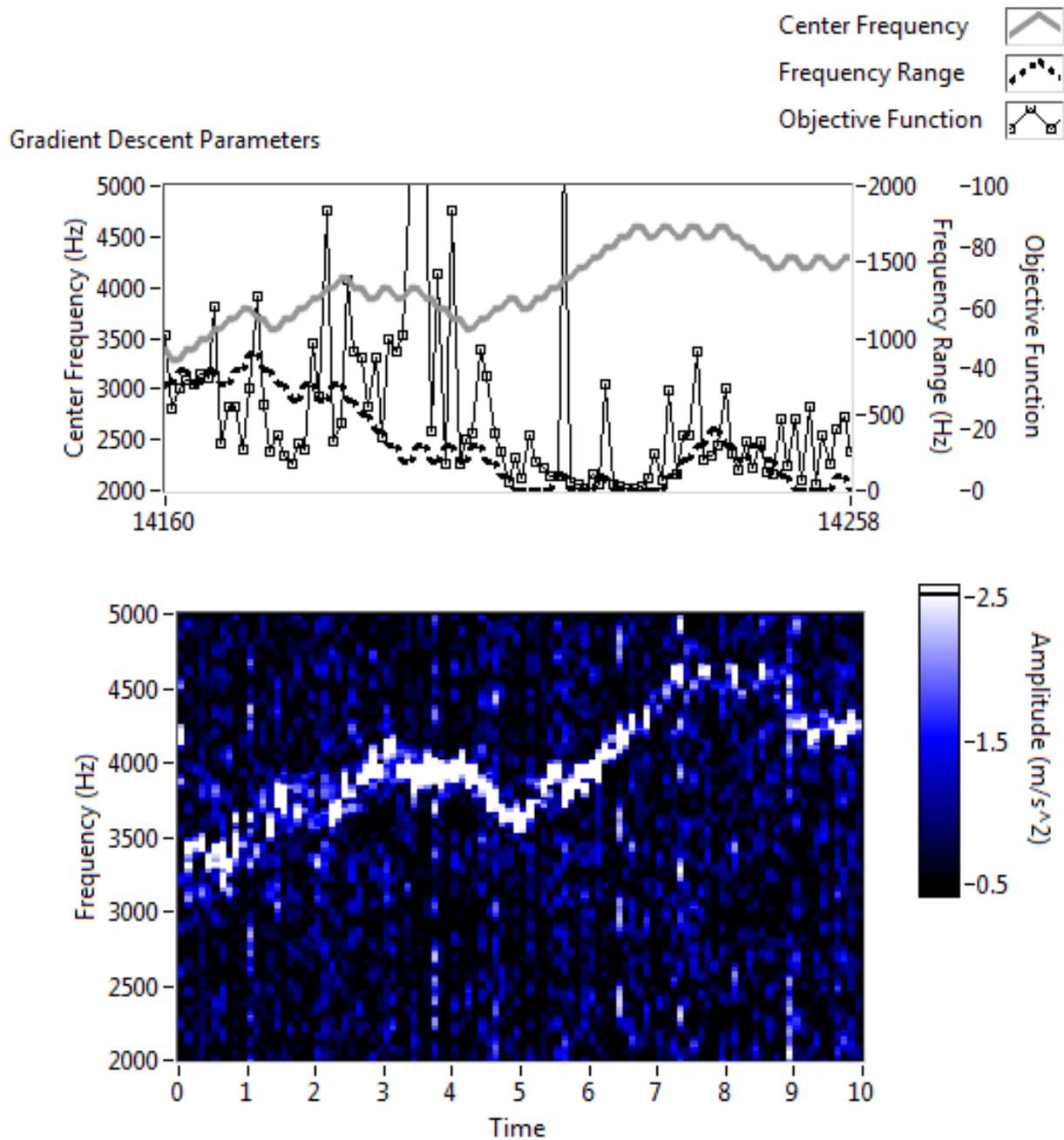


Figure 47. ARPWM over time at 20 Hz fundamental.

Figure 49 is similar to Figure 46 and Figure 44, however our iterative mapping of the search space has been expanded to include distribution ranges up to 3 kHz.

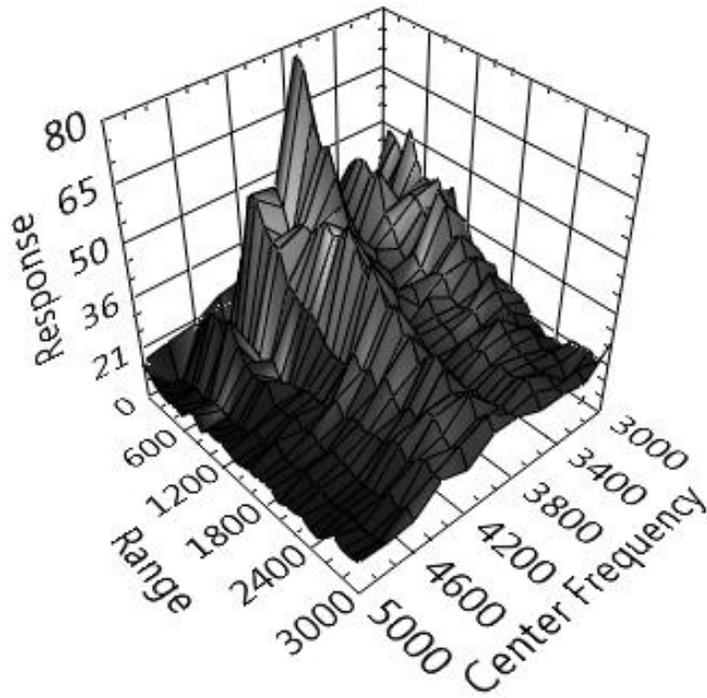


Figure 48. Map of search space generated when the mapping procedure was repeated while allowing the range to increase up to 3 kHz.

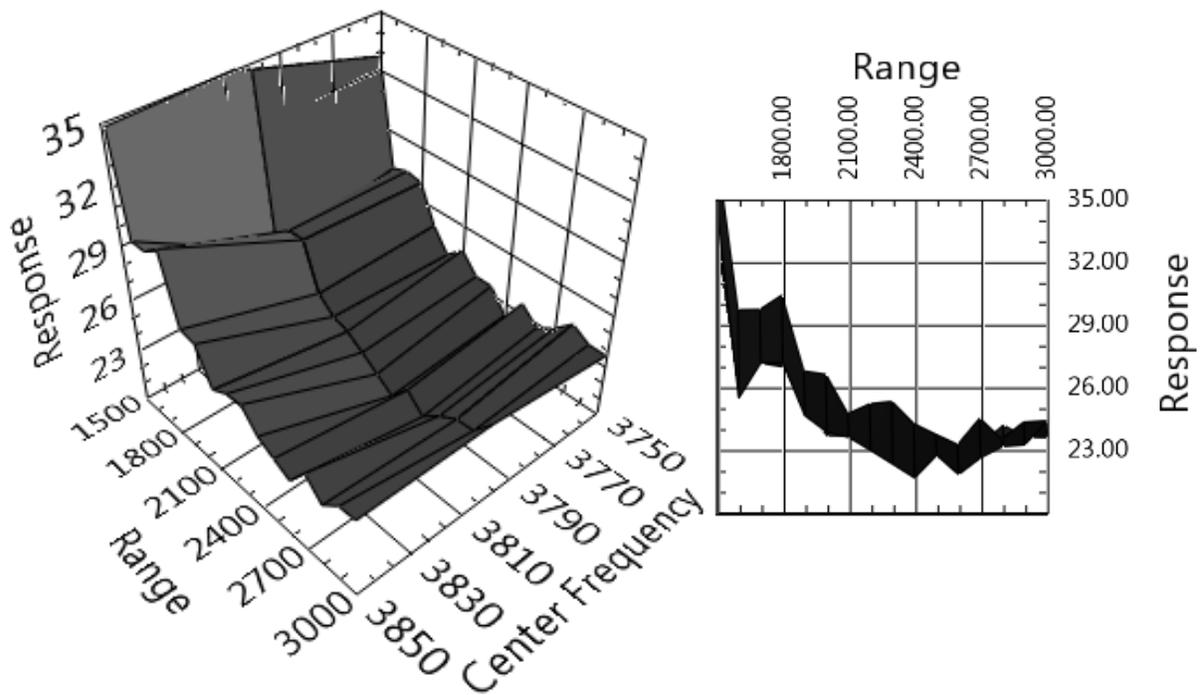


Figure 49. Close-up of system response near a resonant peak (left) and frequency slice response versus range (right).

Figure 49 shows a close-up of a region around one of the resonant frequencies of the system (3900 Hz). Notice that the slice at around 3850 Hz along with other neighboring frequencies actually begins to increase once the range of the switching frequency distribution nears 3 kHz. This is because the distribution of the switching frequency is now overlapping the resonance because it is so wide. Once the range becomes large enough (such that the switching frequency distribution overlaps with a system mechanical resonance) increasing the range further results in an increase in objectionable vibration.

In order to demonstrate the behavior of ARPWM configured with a constant learning rate (found through trial-and-error), tests were performed starting at different initial switching frequency distributions. Figure 50, Figure 51, and Figure 52 show the time-frequency behavior in 3 different cases. Figure 50 demonstrates a best case scenario; Figure 51 demonstrates a moderate case scenario; and Figure 52 demonstrates a worst case scenario.

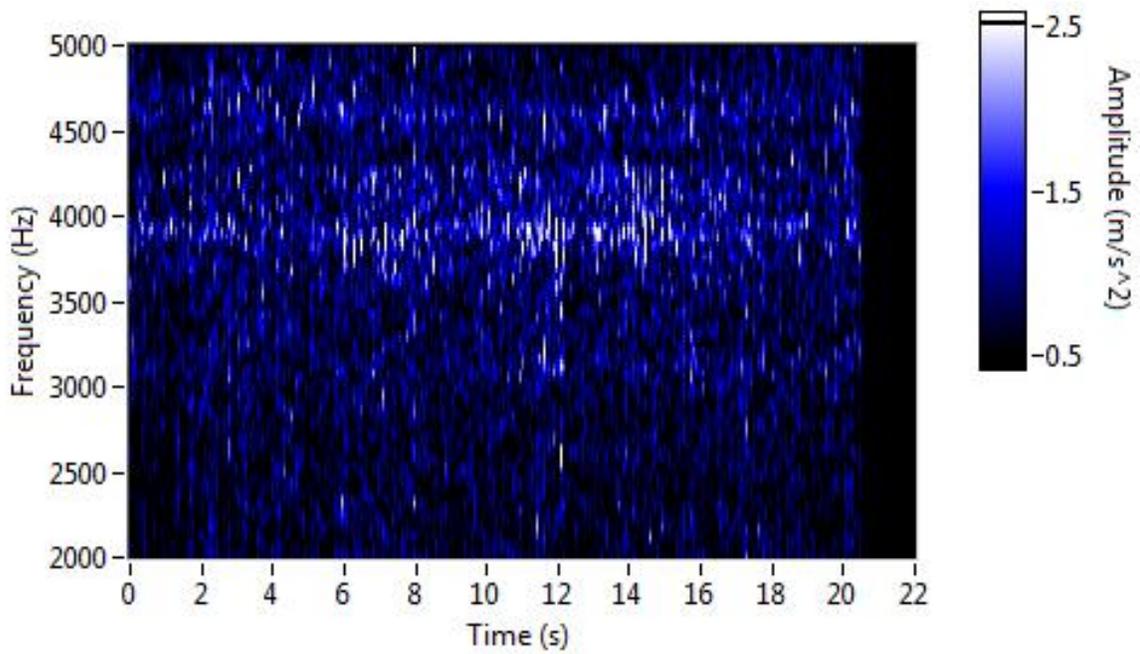
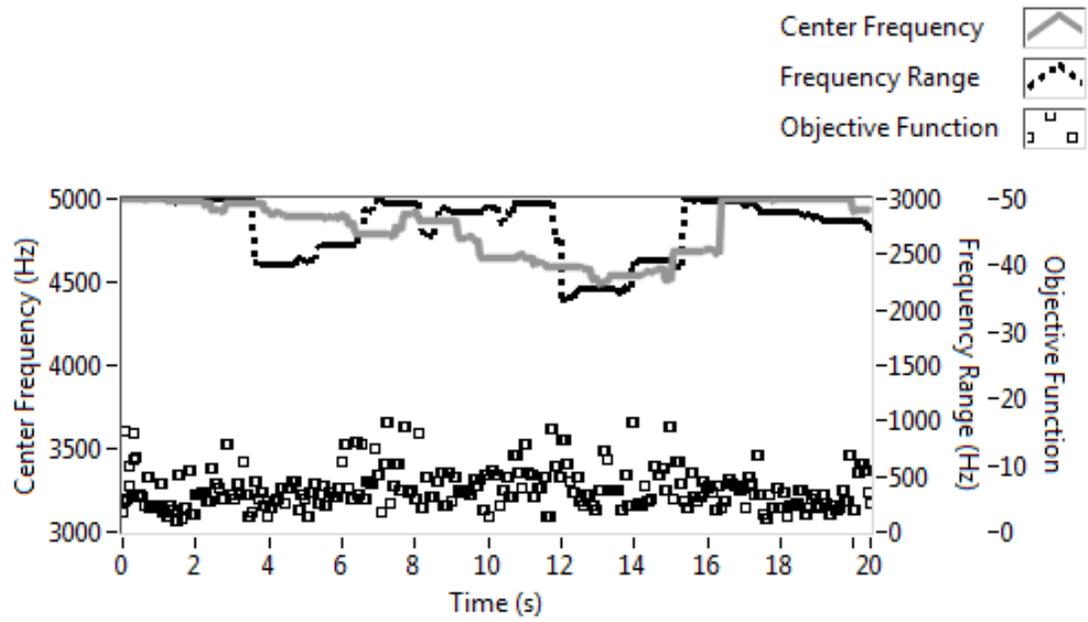


Figure 50. Best-case initial carrier frequency distribution conditions. Since the initial conditions were near-optimal in a smooth region of the search space, there wasn't much searching to do.

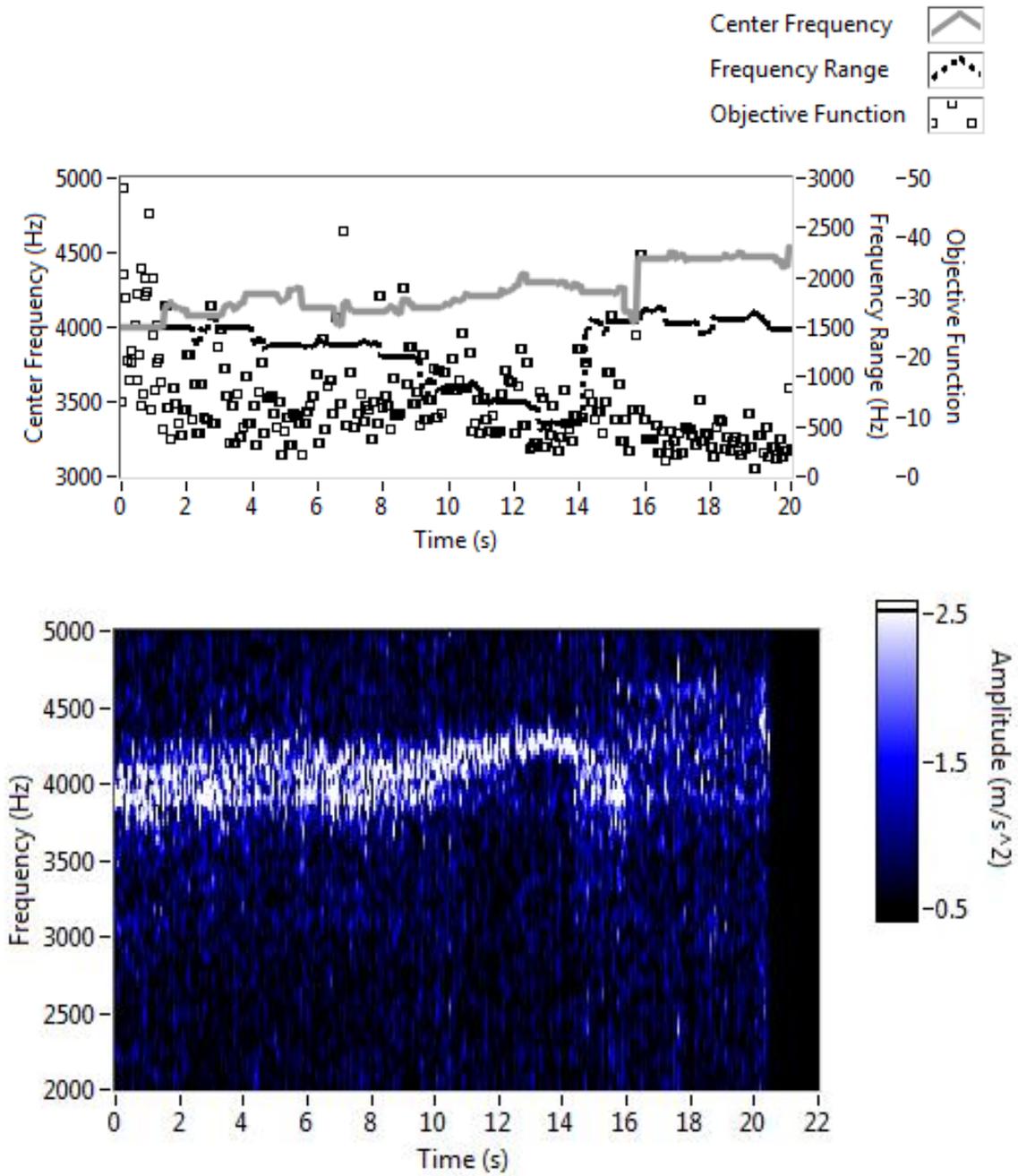


Figure 51. Moderate-case initial carrier frequency distribution conditions. Notice how the range decreases over the first 13 seconds; the search space is not so trivial in this region and the gradient descent algorithm is misled for a while.

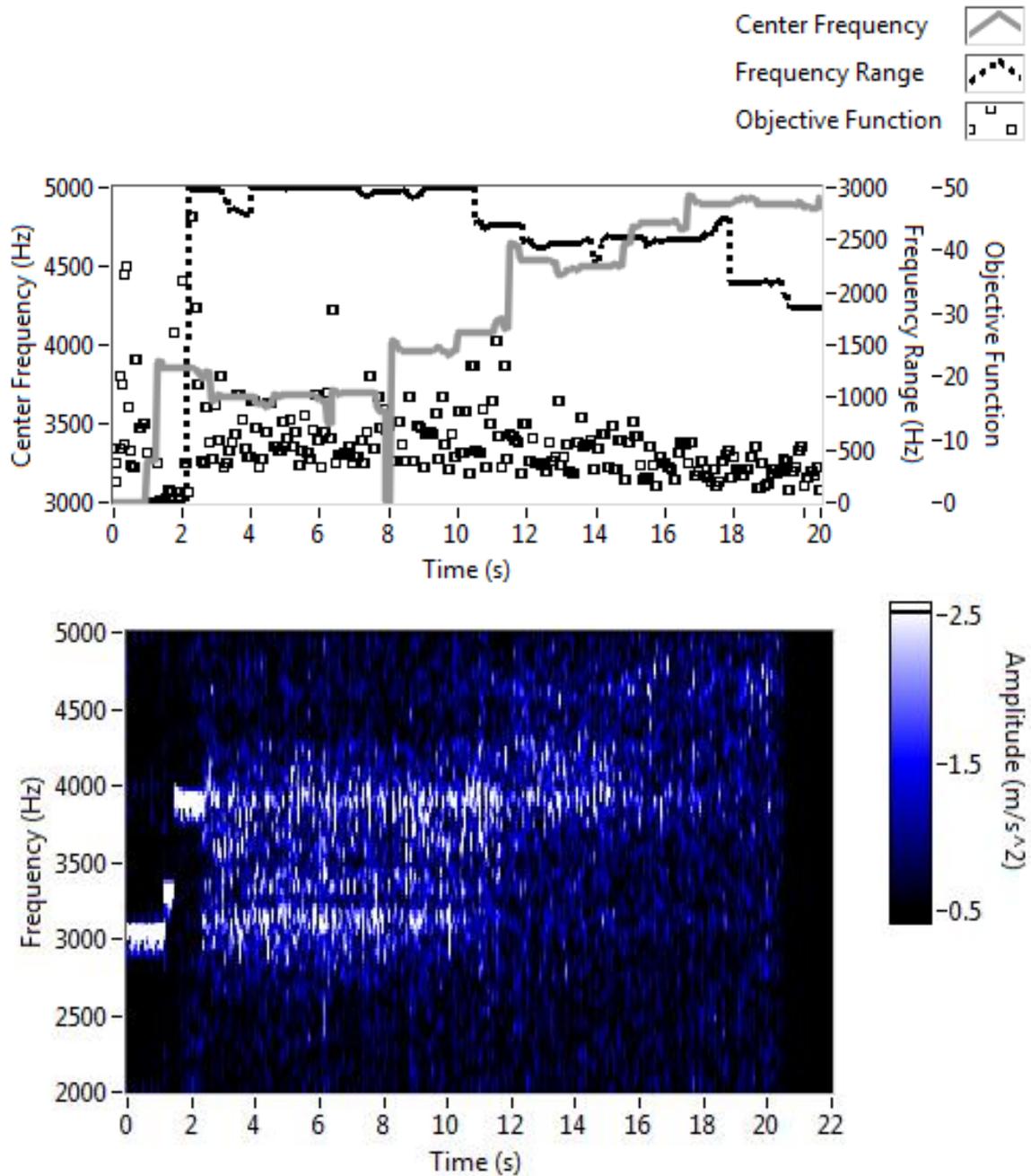


Figure 52. Worst-case initial carrier frequency distribution conditions.

Notice how the gradient descent algorithm is able to make it through the local minimum between the resonant peaks at 3000 and 4000 Hz in order to find the global minimum region. This behavior is dependent on parameters such as the learning rate. If the search space were

more deceptive, some other hill-climbing technique employing a type of momentum would become necessary over just a basic learning rate parameter (or a search technique less susceptible to local minima could be employed). Alternatively, we can explore the previously discussed variable learning rate approach; with a variable learning rate, the gradient descent algorithm is more accommodating to complicated search spaces. Figure 53, Figure 54, and Figure 55 compare temporal behavior at a couple different constant learning rates with the variable learning rate ARPWM algorithm. In some cases, the variable learning rate algorithm takes a while to settle into the correct peak learning rate value, but does not require manual parameterization.

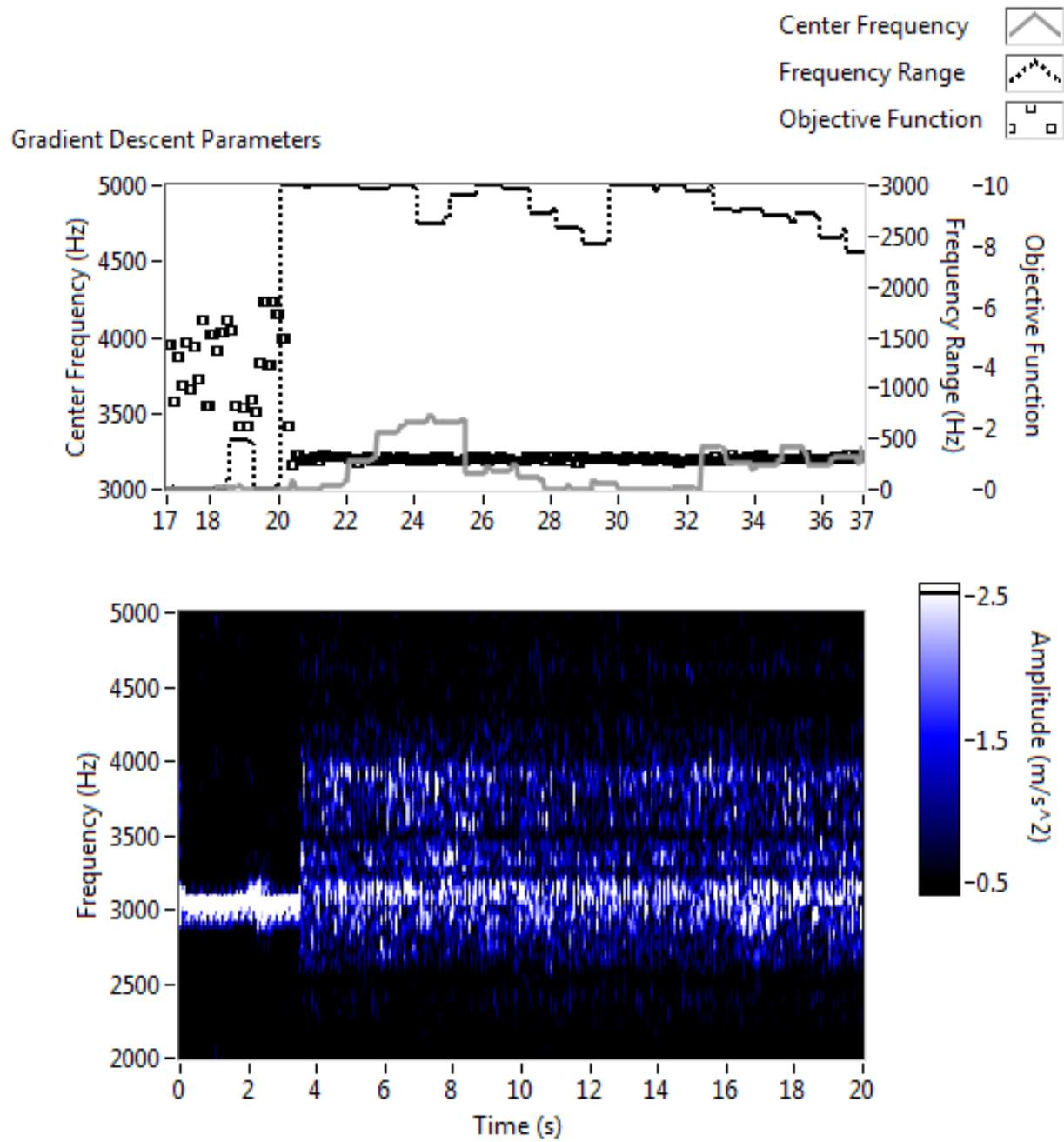


Figure 53. Temporal behavior of ARPWM with a constant learning rate of 0.01.

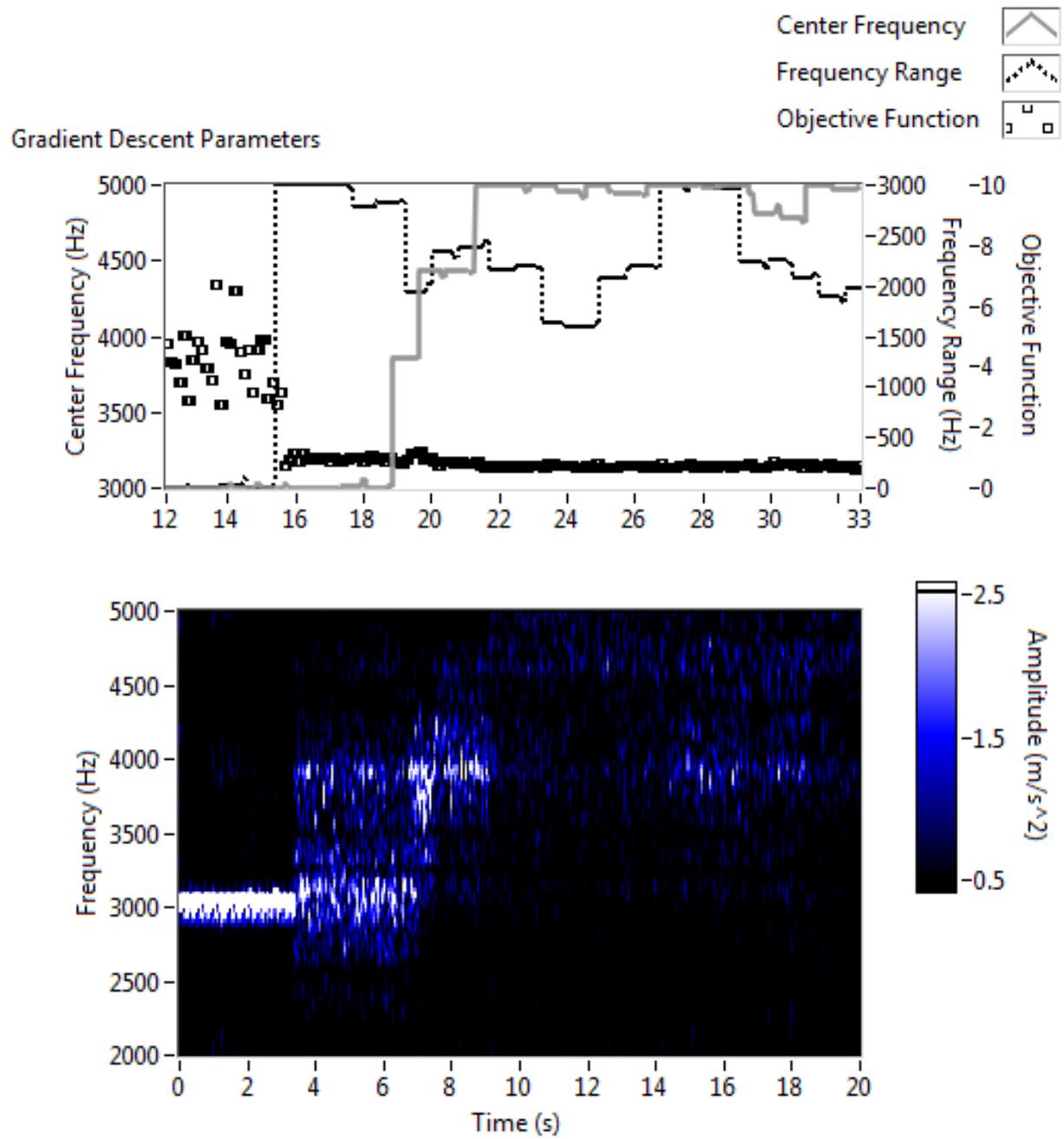


Figure 54. Temporal behavior of ARPWM with a constant learning rate of 0.02.

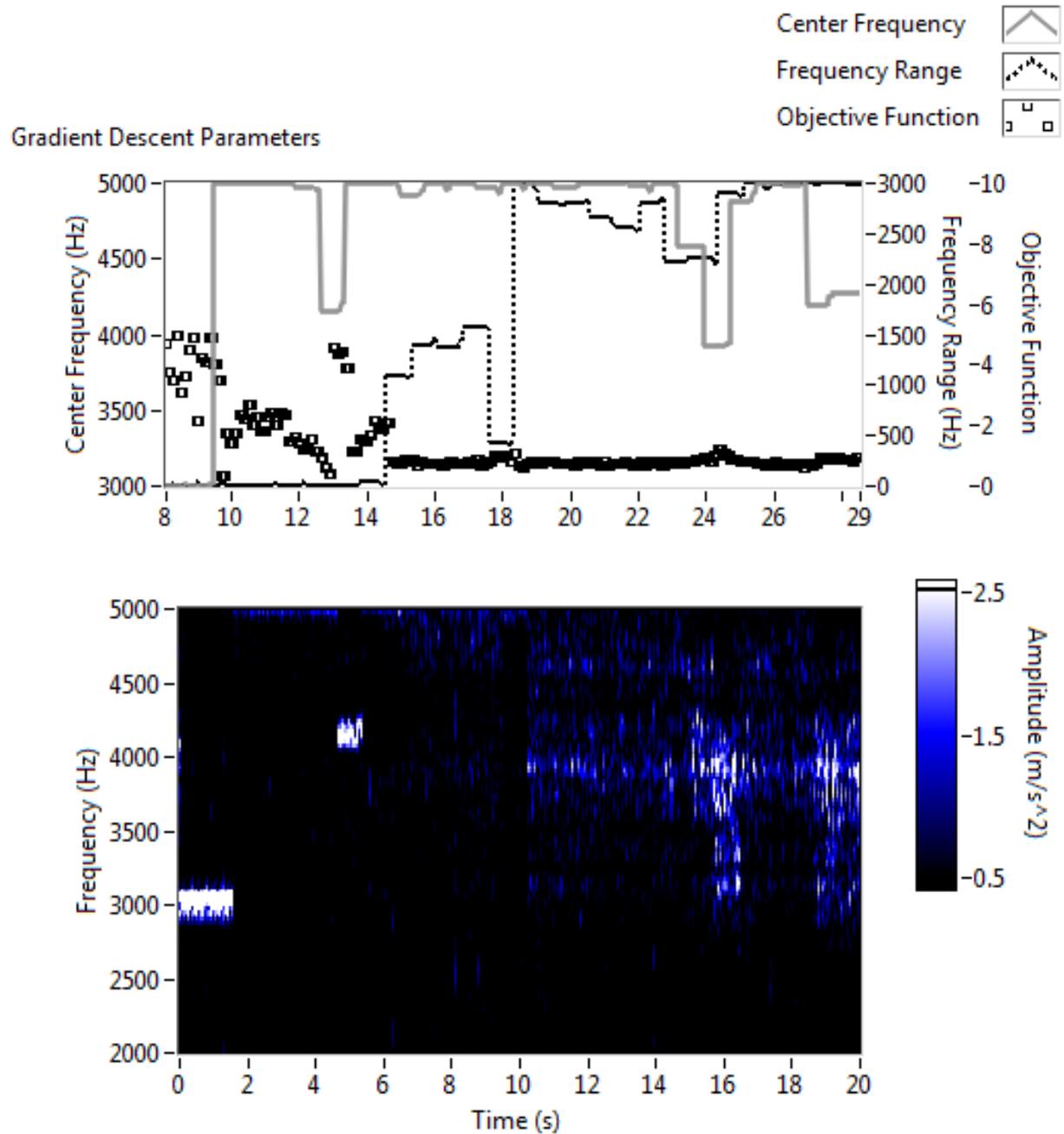
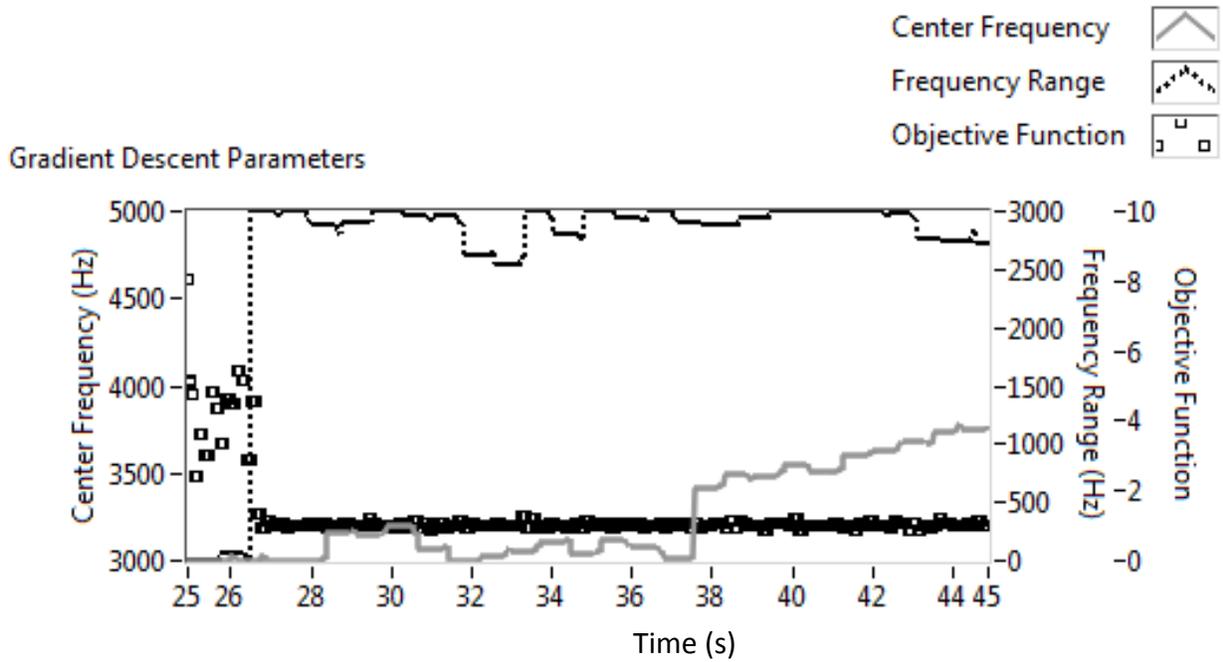
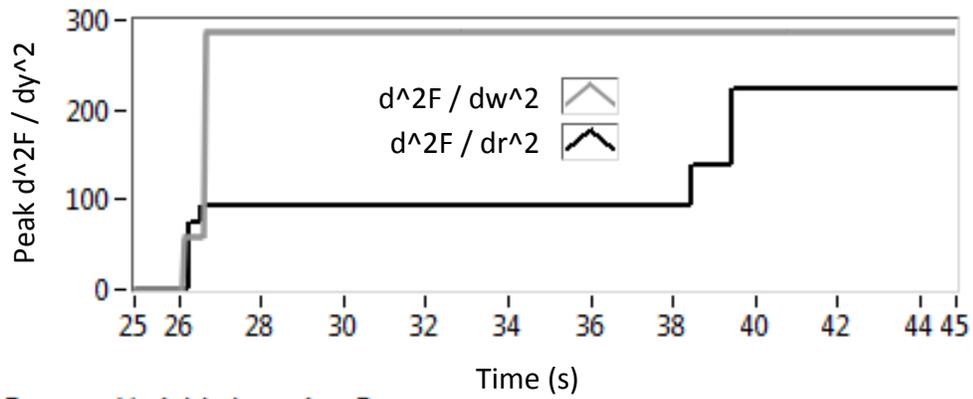


Figure 55. Temporal behavior of ARPWM with the variable learning rate algorithm

In order to illustrate the parameters of the variable learning rate algorithm over time, a couple different instances are included in Figure 56 and Figure 57.



Objective Metric: Peak 2nd Derivative



Gradient Descent Variable Learning Rate

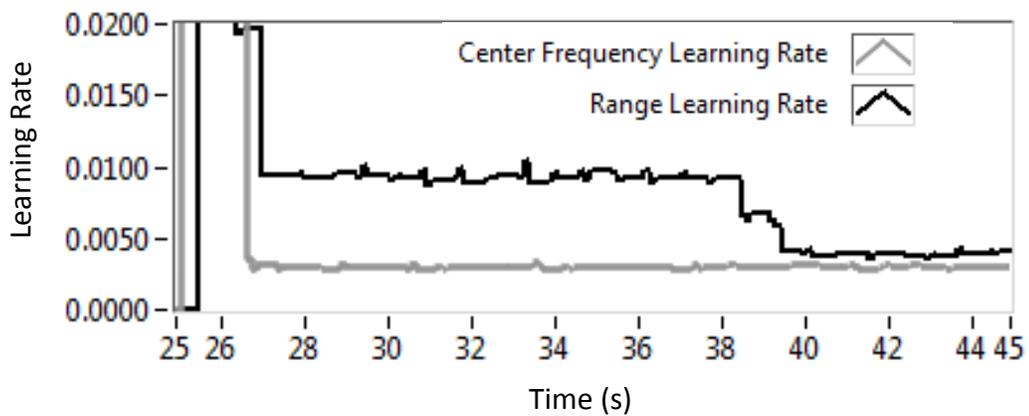


Figure 56. Variable learning rate metrics over time during execution of the variable learning rate algorithm.

Figure 56 illustrates how the peak learning rate is a function of the maximum encountered second derivative of the objective metric with respect to the gradient descent independent variable. It is very clear that the peak learning rate (for the gradient descent algorithm responsible for varying the distribution range) decreases when a new maximum second derivative of the objective metric is encountered.

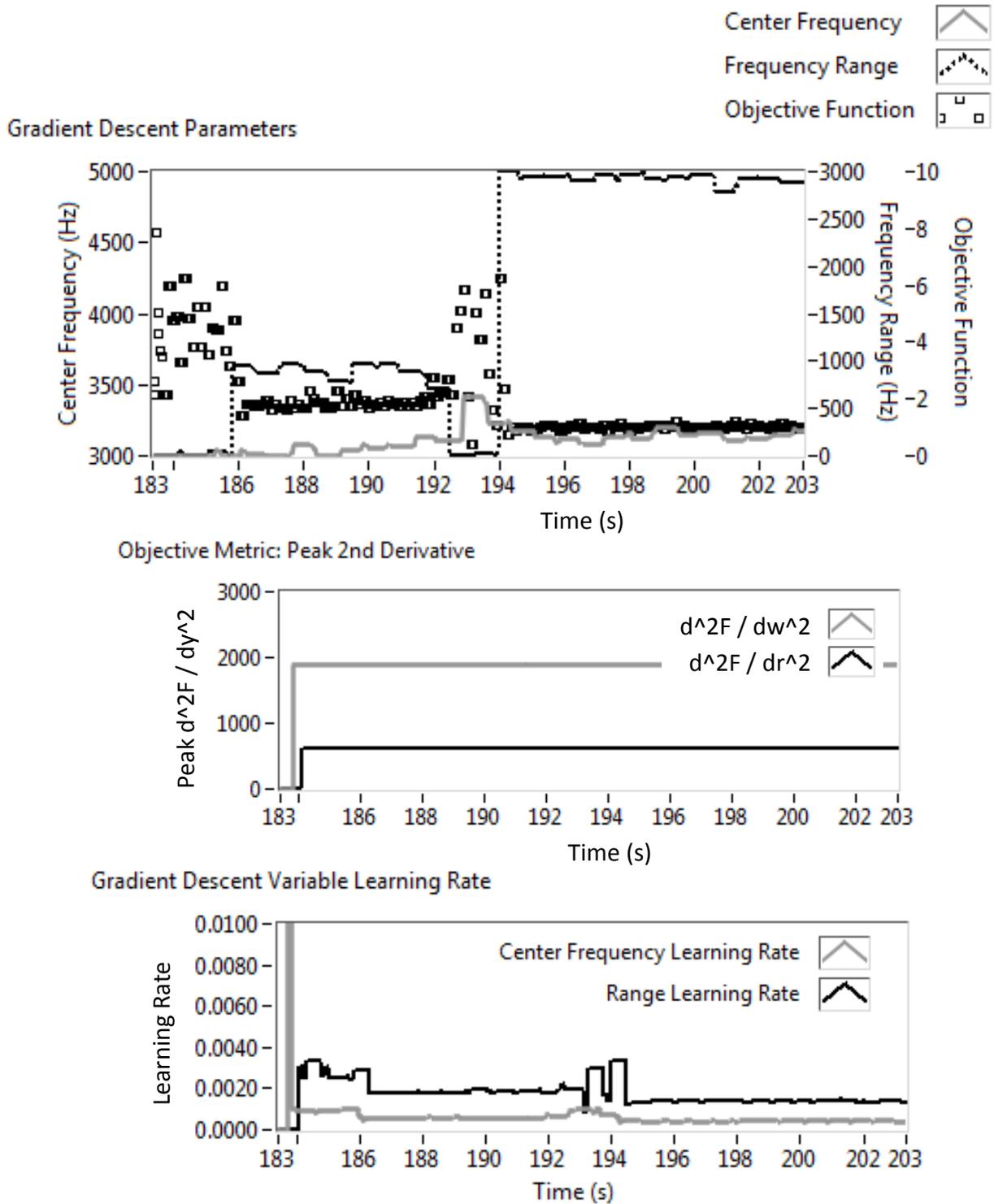


Figure 57. Variable learning rate metrics over time during another execution of the variable learning rate algorithm.

The example in Figure 57 illustrates a few things. First, just as in our analysis of the representative signatures, the peak second derivative of the objective metric with respect to the independent variable is much higher for the center frequency than it is for the range. Additionally, and consequentially, the peak learning rate for the center frequency is much less than that of the range. One can also observe the increase in instantaneous learning rate in the time range over which the response is momentarily higher (which allows the escape from a local minimum).

Conclusions

ARPWM offers a good option for reducing objectionable vibration and sound in electric power conversion applications in which system response is unknown or variable. The gradient descent approach is a very simple and effective approach for providing the desired adaptive behavior. This approach could be further refined, since there are several system parameters that would allow for a wide range of dynamic behavior.

Hill climbing algorithms other than gradient descent may be worth investigation, especially since as the gradients point nearly orthogonally to the shortest direction to a minimum point, gradient descent increasingly 'zigzags' [34]. Additionally, asymptotic rate of convergence is inferior to many other methods [34]. It might also be worth investigating GA or PSO search in place of gradient descent to improve convergence to global minima over local minima; the only disadvantage of such a technique that would need to be overcome would be the unwanted excitation of resonant frequencies encountered with bad solutions while searching for good solutions.

In most cases, a properly selected constant learning rate will probably perform sufficiently. In order to be more robust, the variable learning rate algorithm could be improved with a 'sliding' maximum response and second derivative calculation – in the event that the system response changes so much that a previous maximum value may keep the learning rate too low for following conditions. We don't have a guaranteed global minimization (just local), but based on the shape of the search space, even a local minimum is usually okay. Again, if a

system were encountered which posed more difficult for global minima convergence, evolutionary search could always be employed; this would also eliminate the need for numerical calculation of derivatives for lower noise susceptibility. Unlike the experiment performed in this thesis, an actual implementation might have a much simpler search space and not as sensitive to learning rate, since the allowable distribution range and center frequency span could be tightened if more is known about a system.

The general purpose of this thesis was to build on existing RPWM techniques in order to improve behavior in unidentified systems or in systems in which the possible mechanical response is complex and changes under various electrical or mechanical conditions. The improvement in the 1kHz to 5kHz range may be marginal if the RPWM center frequency is thoughtfully placed in a system whose response does not vary significantly across applications and operating conditions, however the additional cost of implementing the gradient descent algorithm is almost negligible for many applications, since it can be implemented on the existing hardware in many applications without increasing software resource demands significantly, and is therefore favorable for applications where system response variation is non-negligible across all operating conditions.

APPENDICES

Appendix A: Arbitrary Motor Control Algorithm Prototyping Hardware

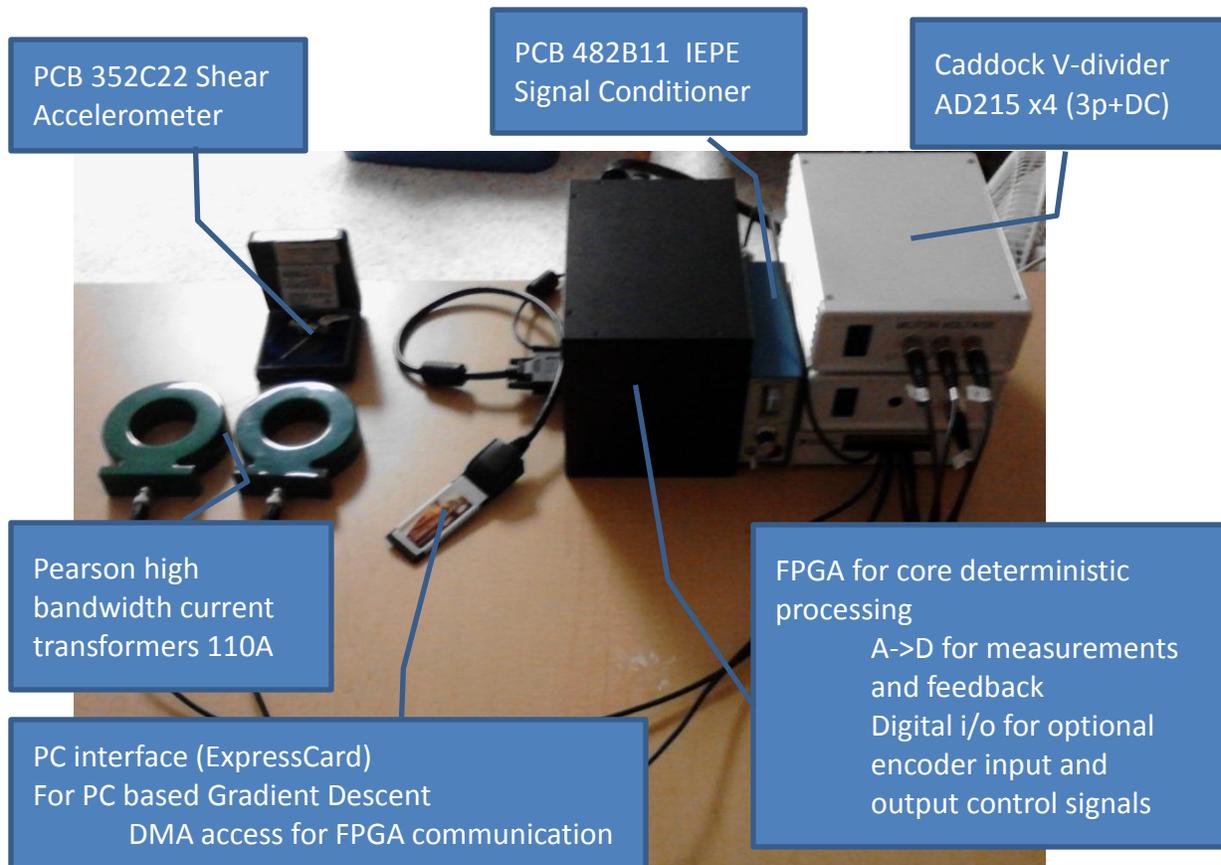


Figure 58. The measurement and control system designed to provide a control algorithm prototyping system.

The control algorithm prototyping system (Figure 58) utilizes a National Instruments[™] 7831R FPGA board with analog and digital i/o. It has a 1M Gate Virtex-II FPGA, 8 Simultaneously Sampled analog inputs (16-Bit, differential, 200kS/s, +/-10V), and 96 Bidirectional DIO lines (TTL-compatible, up to 40MHz).

In order to be able to measure the high phase voltages, a signal conditioning circuit (Figure 59) was designed to divide and isolate the incoming voltages. Some Analog Devices[™] AD215BY isolation amplifiers were used with Caddock[™] 1776-C681 decade voltage dividers. The AD215BY provides low distortion, 1500V RMS galvanic isolation, >100kHz Bandwidth, -80dB harmonic distortion @ 1kHz, and +/- 0.005% Non-linearity. The Caddock precision decade resistor voltage divider is 10Mohm, with 0.05% accuracy, and offers dividing ratios from 10:1 to 10,000:1.



Figure 59. The voltage signal conditioning circuit.

For phase current measurements, two Pearsontm current monitors (Model 110) were selected. They provide an output ratio of 0.1Volts/Ampere, are rated for 65 Amps RMS (5000A Peak), and have a wide ~1Hz-20MHz bandwidth.

For vibration feedback a PCB 352C22 single-axis accelerometer was selected with ~10mV/g sensitivity, 0.3-20kHz (+/-3dB) bandwidth, and +/- 500 g peak input range.

An Optocoupler circuit (Figure 60) was assembled in order to provide isolation for the digital control signals. Fairchildtm FOD260L optocouplers were selected for high speed switching characteristics. The optocouplers were just breadboard mounted due to time constraints.

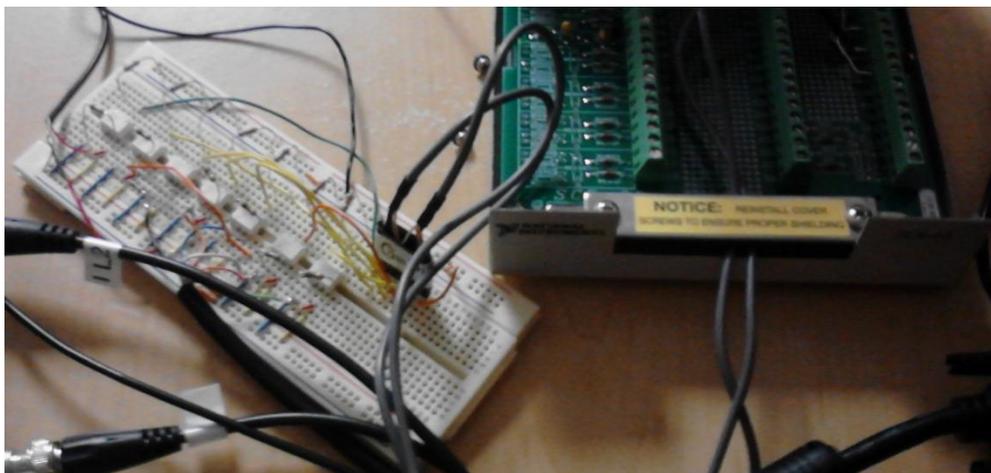


Figure 60. Optocoupler circuit for digital signal isolation.

A design reference inverter (Figure 61) from STMicroelectronicstm was selected to drive the motor. Model UM0900 (1 kW 3-phase motor control demonstration board featuring IGBT intelligent power module STGIPL14K60 STEVAL-IHM025V1) was selected because it was design to accept external PWM switching control signals.

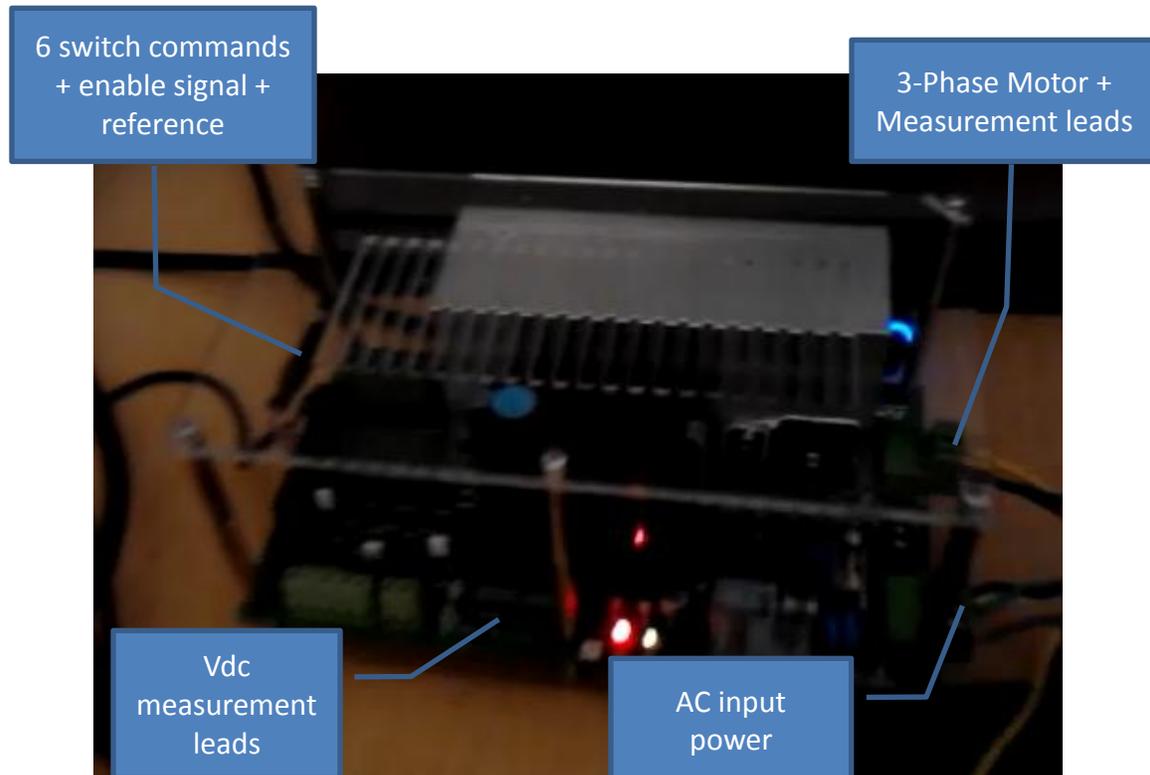


Figure 61. Reference design motor drive.

The motor used in experiments was a Marathontm D390 (1/3 HP, 208-230 / 460Vac, 1.7-1.8 / 0.9A).

Appendix B: Software Details

FPGA Code

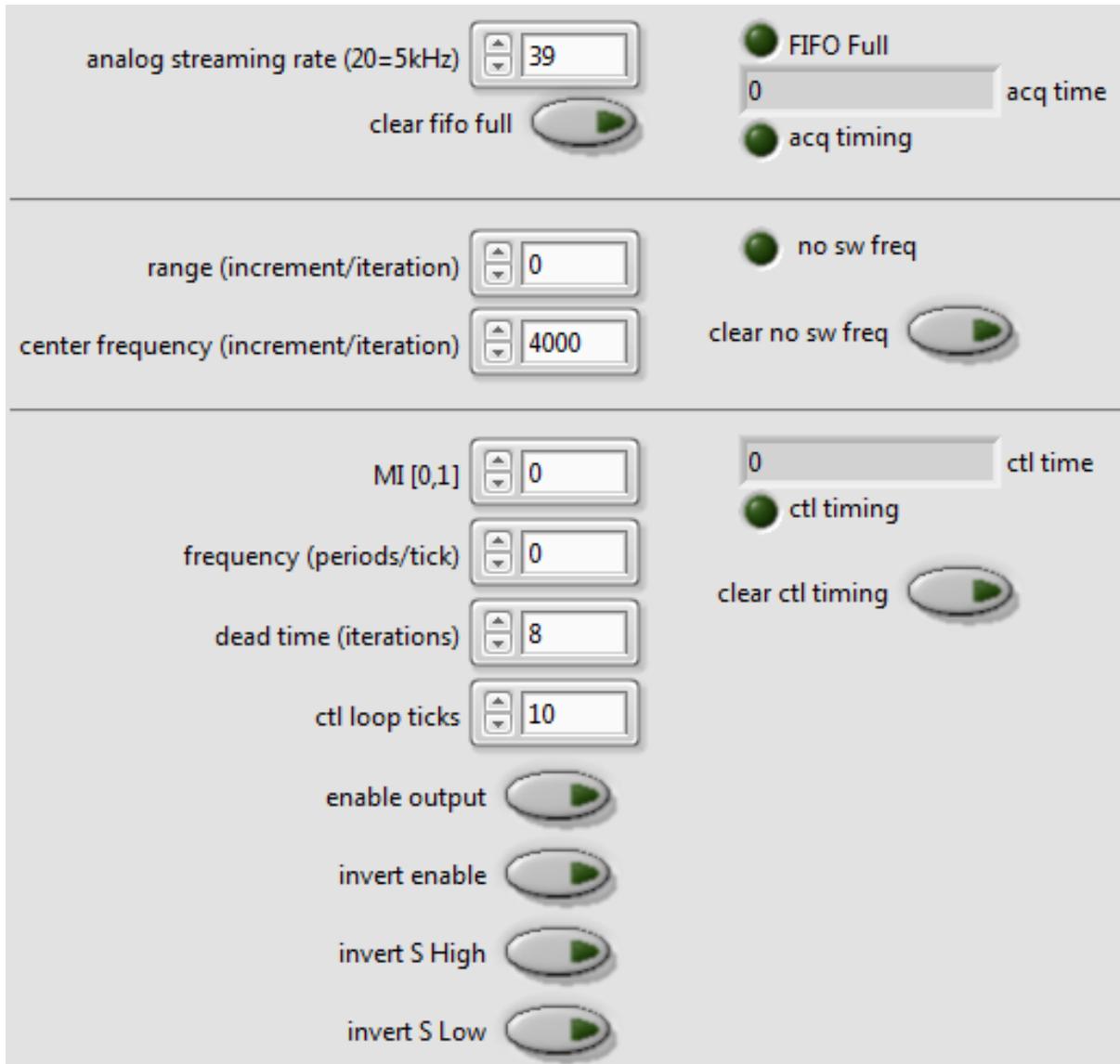


Figure 62. Screenshot of LabVIEW[™] FPGA code Front Panel

The host software can communicate with the FPGA system through DMA channels and through references to the front panel objects depicted in Figure 62. The system was designed to allow connections to a variety of drives by allowing the inversion of the control signals. Additionally, the analog sampling rate, control loop rate, and deadtime parameters are all configurable.

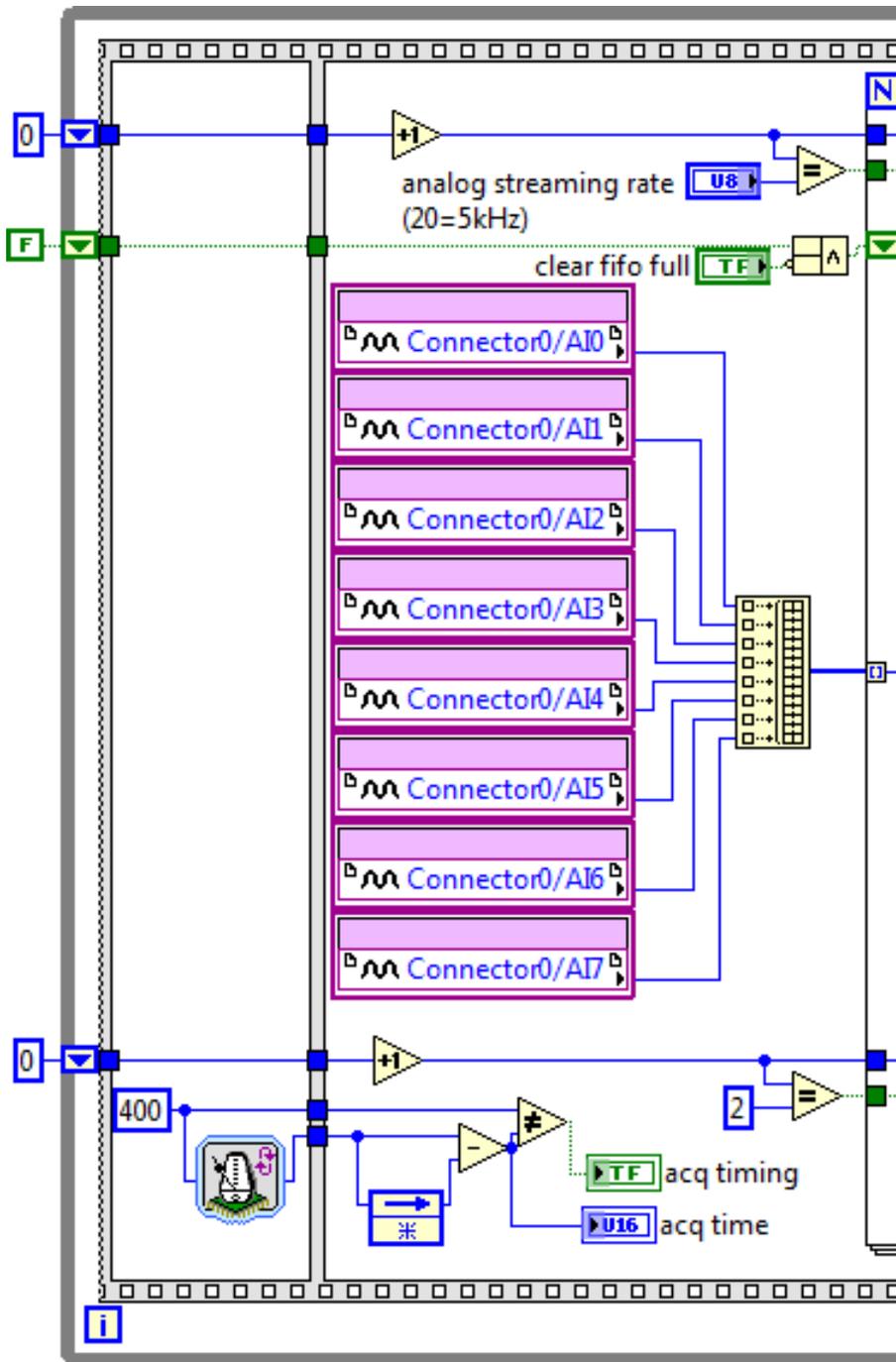
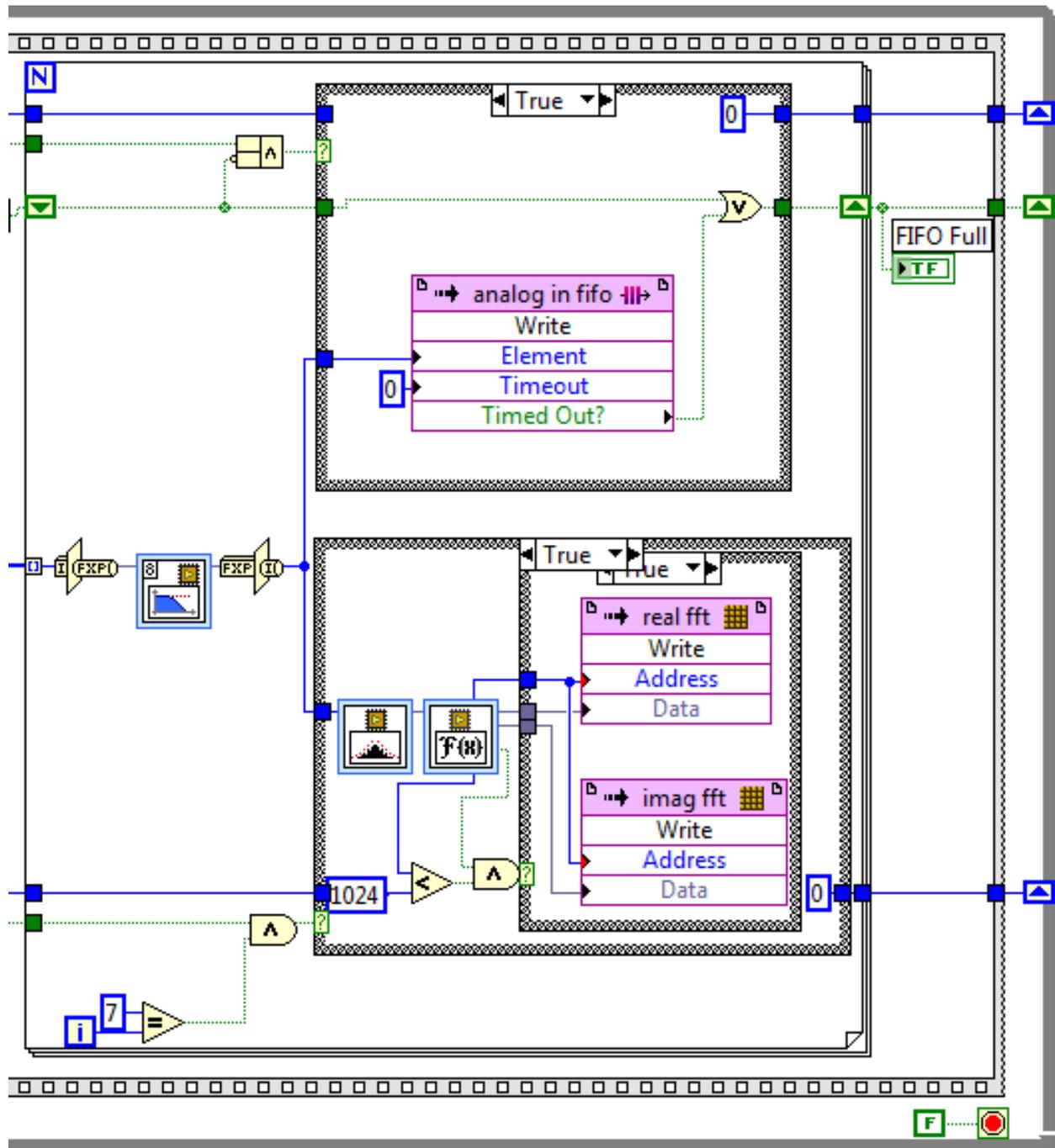


Figure 63. Screenshot of FPGA Acquisition, Filtering, Downsampling, FFT, DMA and Storage Code

Figure 63 (cont'd)



The analog input thread is pictured in Figure 63. It employs a 100 kS/s sampling rate (increasable to 200 kS/s if desired) for acquisition of the various system voltages, currents and vibration. A 20 kHz lowpass filter is applied to the incoming data in order to eliminate aliasing artifacts from higher frequency content before down-sampling to 20kS/s. A point-by-point flattop window algorithm is applied to 2048 sample frames to reduce spectral leakage before it is passed to the point-by-point FFT transformation. The frequency bins are then updated in a round-robin fashion in a circular memory buffer. The entire spectrum can be transferred on-demand using DMA to the host program (Figure 64). The filtered analog measurements can also be streamed to the host.

Overlapping FFTs could be used to increase time resolution. This could be implemented in point-by-point algorithms by using multiple instances of the FFT algorithm, where each instance is initialized at different phase offsets. For example if a 50% overlap were desired, two instances would be used; they would be offset by a 1/2 frame size. They could even write to the same memory space if we are just using single sided spectrum, since half the time we are just throwing away data anyway.

In some applications, the phase shift delay due to the lowpass IIR could affect behavior. Also, in most applications this filter could be completely eliminated without significant impact, but it was employed here to encompass a worst case scenario of signal propagation delay.

The one-sided spectrum is transferred to the host by reading from the circular memory buffer and writing to the DMA FIFO to the host, on demand (Figure 64).

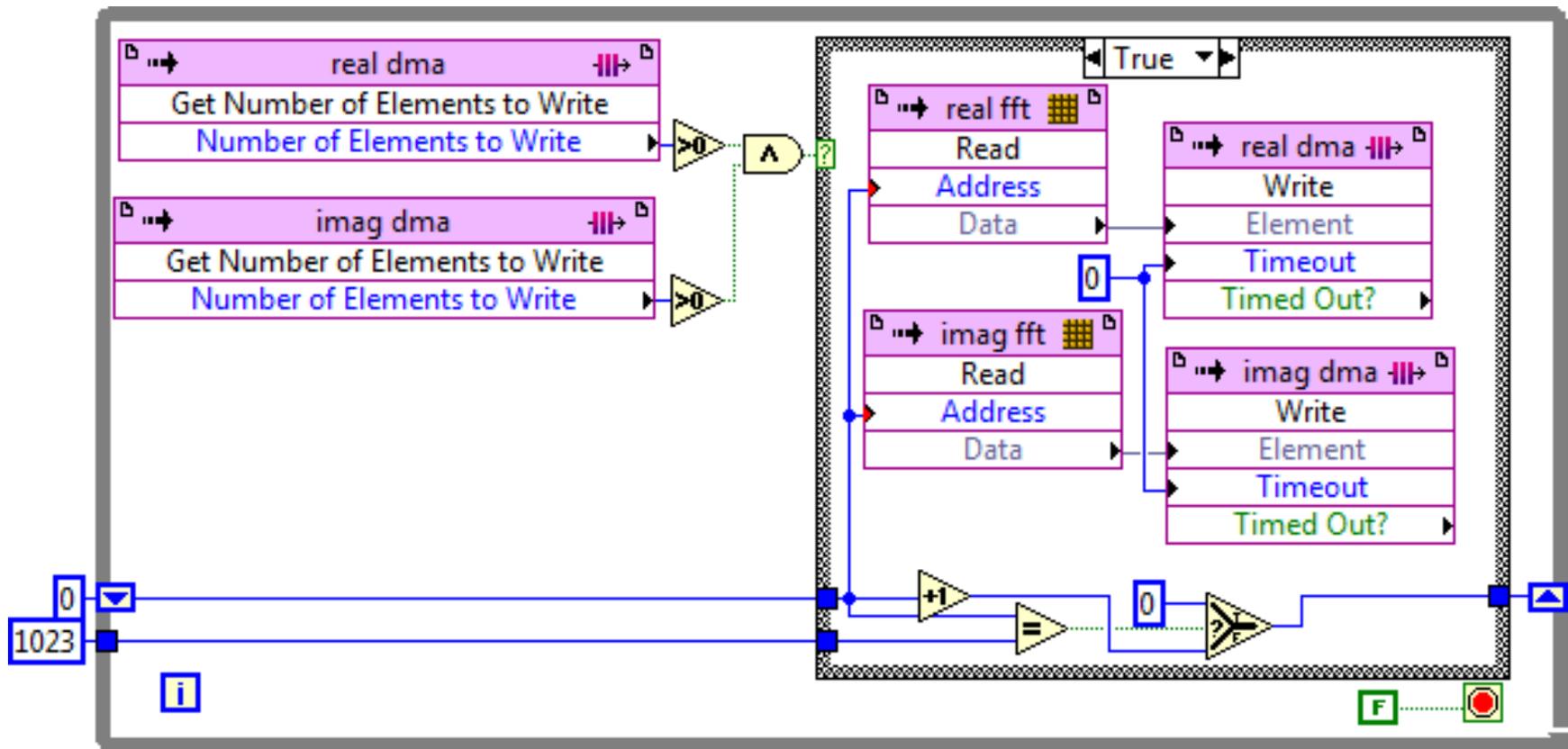


Figure 64. Screenshot of FPGA One-Sided Spectrum Transfer to Host Code

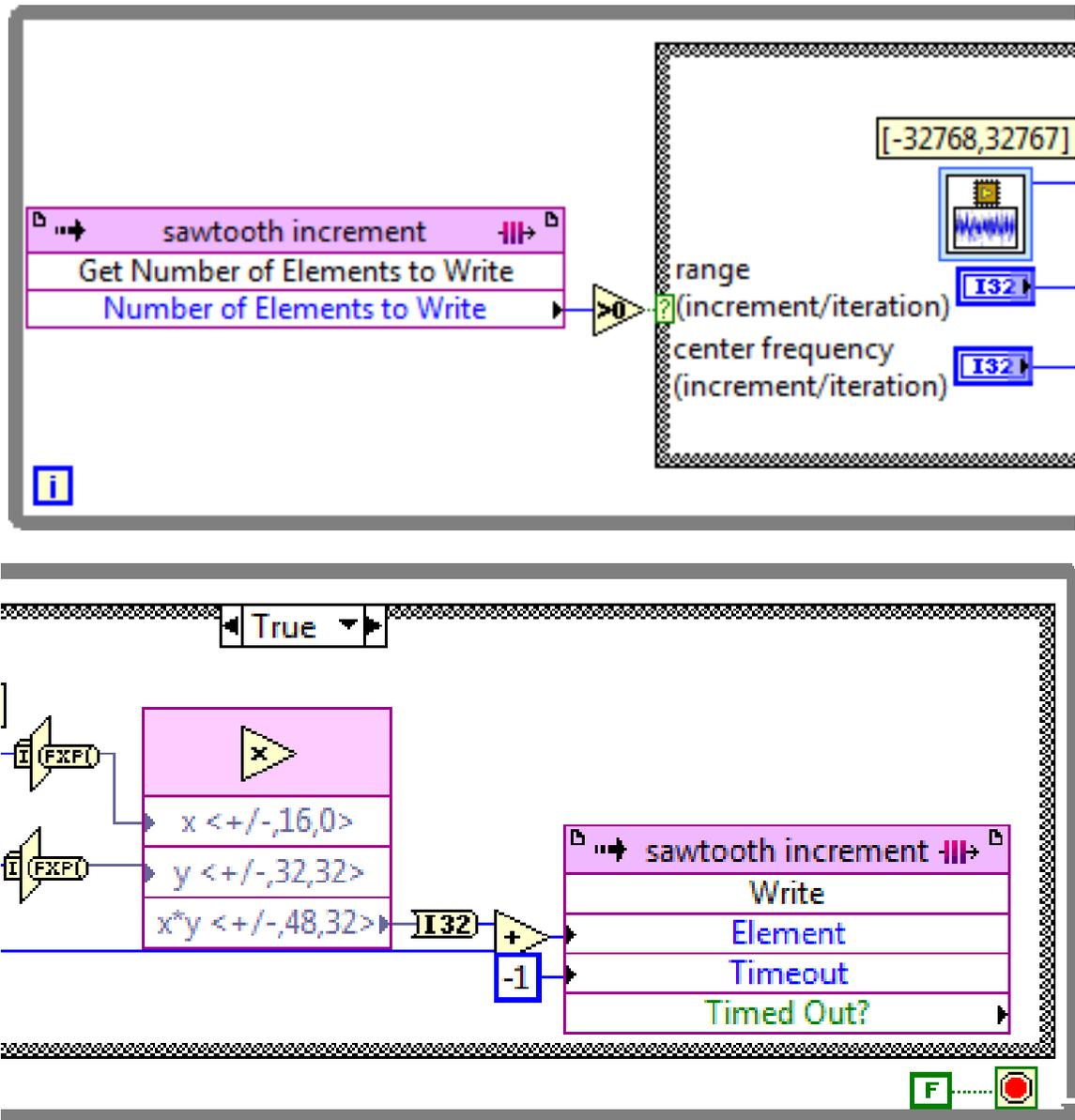


Figure 65. Screenshot of FPGA 'Next Carrier Frequency' Generation Thread

Figure 65 contains the code used to implement the ARPWM Carrier Design/Shaping Thread. It generates the frequency of next carrier period and FIFOs that value to the main control thread (as control thread pulls the value off, this thread pushes a new value on). The main control thread never needs to wait for next frequency value to be generated – it is always available, since it can be generated in far few clock cycles than are occupied by an entire switching period.

Each frequency is selected based on uniformly distributed pseudo-random numbers provided by a white noise generating subroutine. The provided pseudo-random numbers are then shaped into the desired distribution by scaling by the desired distribution range and offsetting by the desired center frequency. The number provided is actually in units of sawtooth signal increment per control loop iteration.

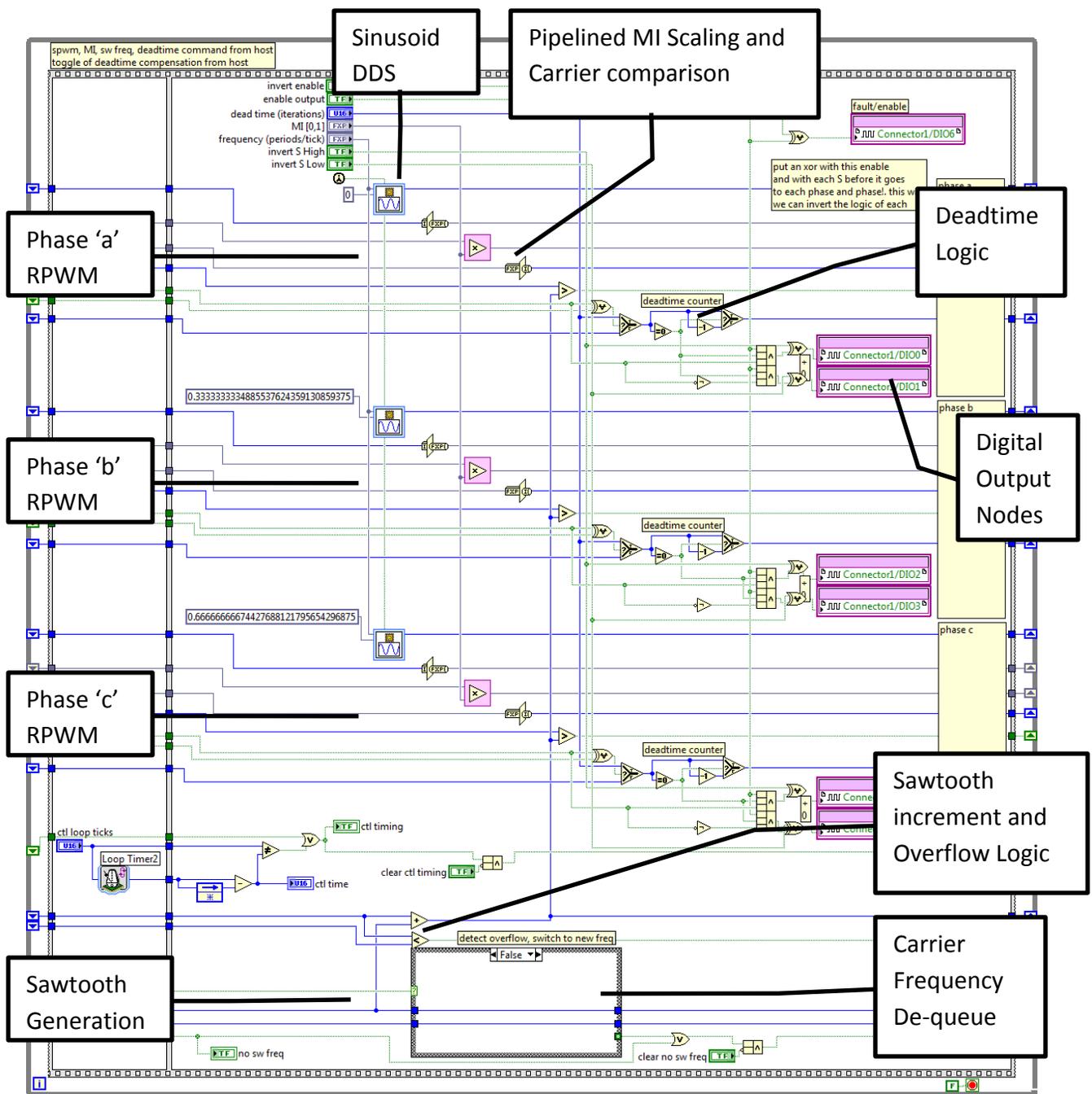


Figure 66. Screenshot of FPGA main RPWM control thread.

The control thread (Figure 66) provides the control signals for RPWM modulation based on the carrier frequencies provided by the carrier frequency generation thread. The reference signals are generated by direct digital synthesis (DDS) of sinusoidal references stored in memory.

Some pipelining is implemented explicitly in the code, while the compiler also automatically pipelines many of the operations in order to be able to achieve fast loop rate (10 clock cycles with 32-bit implementation (4MHz) still with room to improve). With this system, a 40MHz loop rate could be achieved at the expense of more propagation delay, which might not be acceptable if this algorithm were modified for certain types of closed loop control.

Host Code

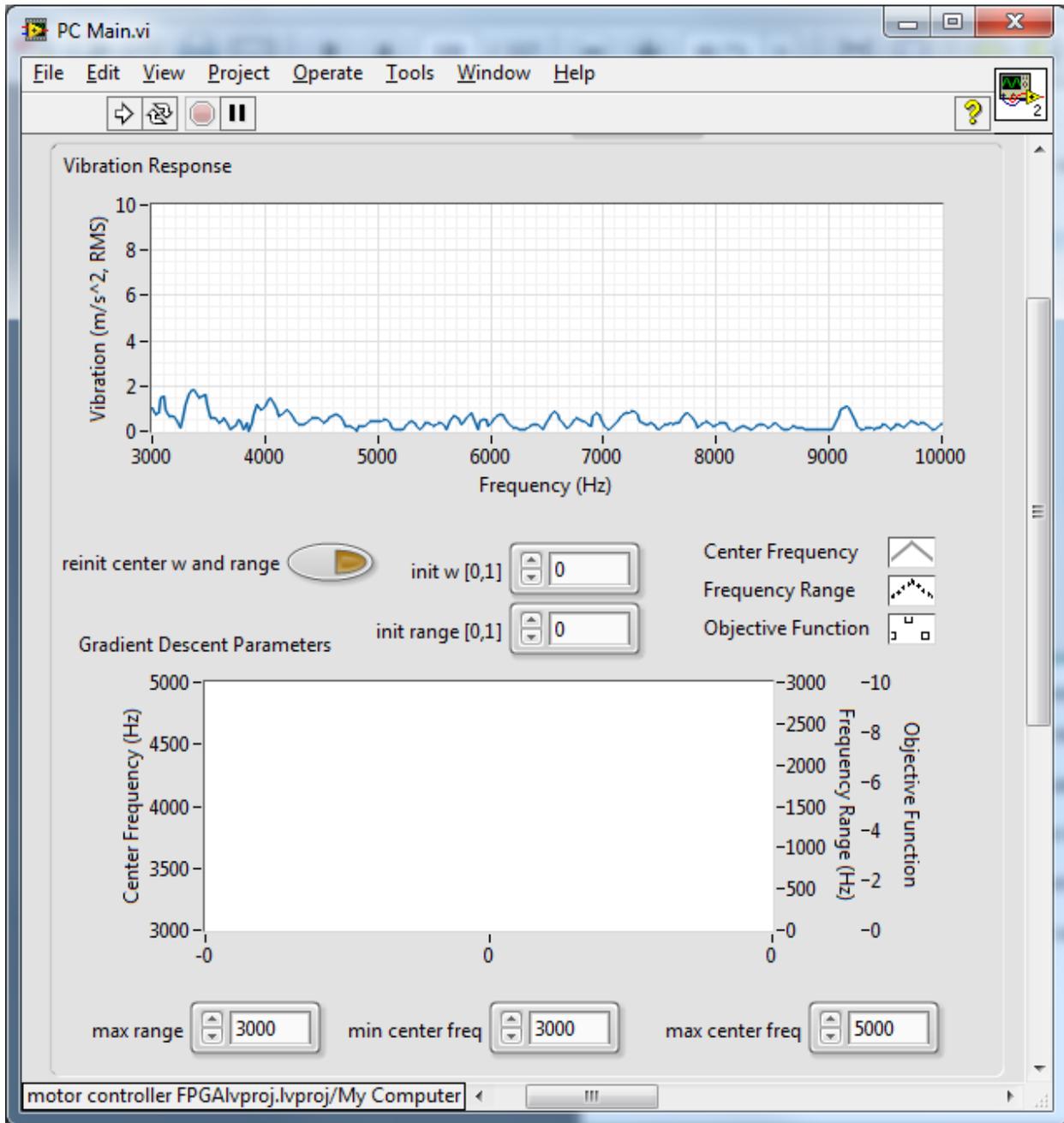


Figure 67. Screenshot of main user interface.

The main user interface (Figure 67) allows the operator to manually vary the fundamental reference signal. The corresponding modulation index is set based on the value entered in the V/Hz control. This interface has graphics to indicate the state of the gradient descent

algorithm, the measured analog signals, and the frequency domain vibration data. Additionally, debug graphics indicating actual loop rates and timeout indicators are included. An automatic sequencing engine was also employed to allow the user to execute a schedule of RPWM distributions in order to map the search space of the system under steady state conditions.

Host Code – Gradient Descent Thread

The entire algorithm could easily fit on FPGA, but due to time-inhibitive compile times and several anticipated modifications and fine tuning during development phases, it was more pragmatic to perform the gradient descent algorithm on a PC program, and just send the description of the desired distribution to the FPGA (as indicated in Figure 62 at the beginning of this section). This thread retrieves the FFT spectrum from FPGA, and calculates the next range or center frequency based on the gradient descent algorithm in Figure 68.

Since most of the processing is offloaded to FPGA, this loop could potentially run up to 1 kHz with no issue, but here we only iterate every 41ms. This is equivalent to one complete update of the spectrum provided by the point-by-point FFT algorithm. Here we are actually averaging two of these spectrums before executing an iteration of the gradient descent algorithm (in order to reduce noise). Also, every ten iterations we are alternating between the two independent variables (center frequency and range), so each of the gradient descent algorithms operate for 5 iterations at a time. In order to promote local search, the gradient descent calculation was slightly modified by adding a small random value to the calculated change in independent variable amplitude.

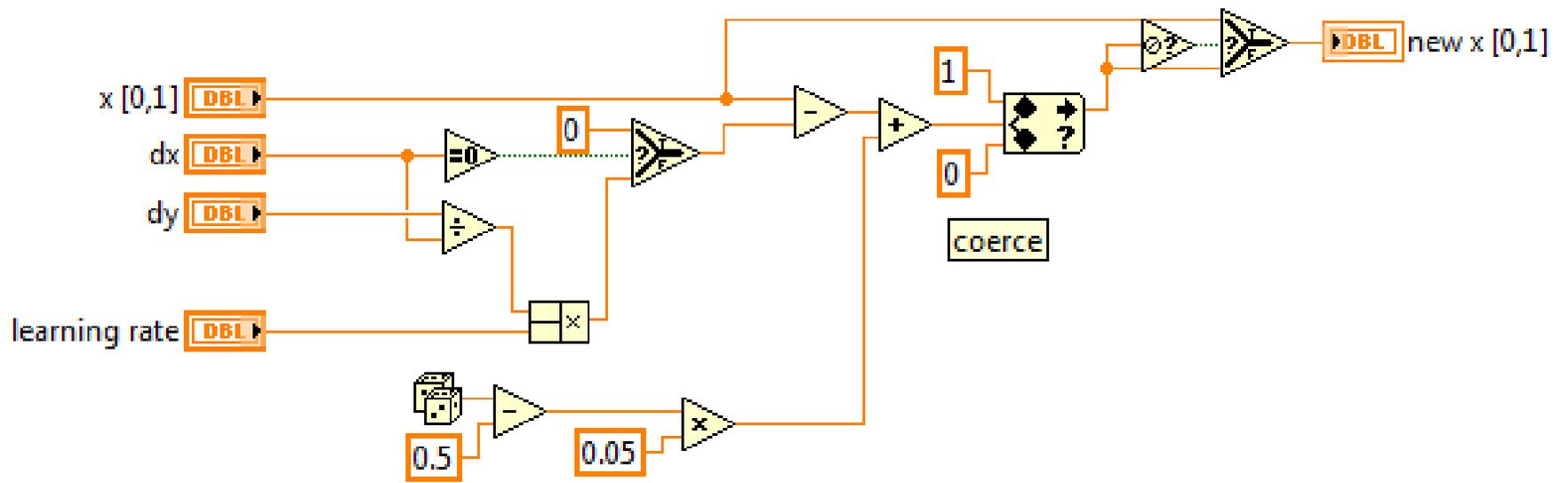


Figure 68. Screenshot of The alternating gradient descent algorithm

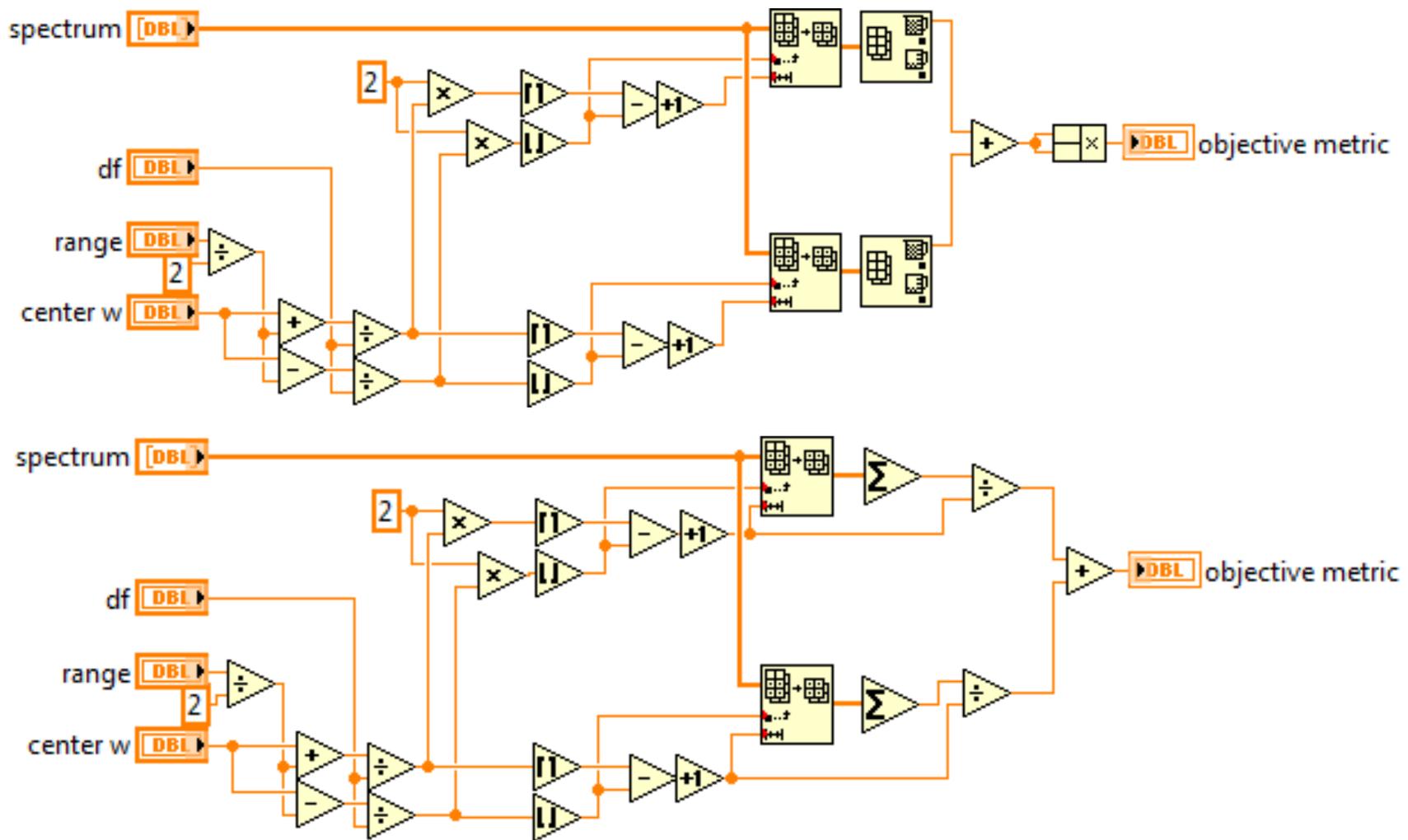


Figure 69. Screenshot of gradient descent objective function for constant learning rate (top) and variable learning rate (bottom).

Figure 69 shows the algorithm for the gradient descent objective function calculation. The algorithm indexes the values in the frequency domain spectrum that lie in the distribution defined by the current modulation frequency distribution (based on the center frequency and range). It also looks at the values that lie in the second order or the modulation distribution. The maximum value (of all of the indexed values) is found for first and second order. The maximum first order value is added to the maximum second order value, and the resulting sum is squared in order to augment the search space into a more aggressive surface. In the case of the variable learning rate objective function, the sum of all values over the ranges of interested is used in place of the maximum occurring values. Additionally, the resulting value is not squared for the variable learning rate function.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] T. Habetler and D. Divan, "Acoustic Noise Reduction in Sinusoidal PWM Drives Using a Randomly Modulated Carrier," in *Power Electronics Specialists Conference, 1989. PESC '89 Record., 20th Annual IEEE* , 1989.
- [2] R. Kirlin, J. Wang and R. Dizaji, "Study on Spectral Analysis and Design for DCDC Conversion Using Random Switching Rate PWM," in *Statistical Signal and Array Processing, 2000. Proceedings of the Tenth IEEE Workshop on* , 2000.
- [3] S. Sathiakumar, Y. Shrivastava and Y.-S. Lee, "Randomized Switching Schemes for DC-DC Power Conversion with Reduced Average Switching Frequency," in *Power System Technology, 2004. PowerCon 2004. 2004 International Conference on*, 2004.
- [4] Y. Shrivastava, S. Sathiakumar and V. Agelidis, "Analysis of 2-Level Random Aperiodic PWM Schemes for DC-DC Converters," in *Power Electronics Specialists Conference, 2008. PESC 2008. IEEE* , 2008.
- [5] A. Stankovic, G. Verghese and D. Perreault, "Randomized Modulation of Power Converters via Markov Chains," *Control Systems Technology, IEEE Transactions on*, Volume: 5 , Issue: 1 , pp. 61-73, 1997.
- [6] A. Stankovic, G. Verghese and D. Perreault, "Analysis and Synthesis of Randomized Modulation Schemes for Power Converters," *Power Electronics, IEEE Transactions on* , Volume: 10 , Issue: 6, pp. 680-693, 1995.
- [7] A. M. Stankovic, *Random Pulse Modulation With Applications to Power Electronic Converters*, 1993.
- [8] D. Trevisan, P. Mattavelli, M. Zigliotto and S. Saggini, "Limited-Pool Random Carrier-Frequency PWM for Digitally Controlled dc-dc Converters," in *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on* , 2006.
- [9] A. Trzynadowski, M. Bech, F. Blaabjerg, J. Pedersen, R. Kirlin and M. Zigliotto, "Optimization of switching frequencies in the limited-pool random space vector PWM technique for inverter-fed drives," in *Applied Power Electronics Conference and Exposition, 1999. APEC '99. Fourteenth Annual* , 1999.
- [10] S. Legowski, J. Bei and A. Trzynadlowski, "Analysis and implementation of a grey-noise PWM technique based on voltage space vectors," in *Applied Power Electronics Conference and Exposition, 1992. APEC '92. Conference Proceedings 1992., Seventh Annual*, 1992.
- [11] L. Ping, L. Zhaoji, X. Fugui and Chenguangju, "Fuzzy Pulse Skip Modulation Mode in DC-DC

- Converter," in *Power Electronics Congress, 2004. CIEP 2004. 9th IEEE International* , 2004.
- [12] A. Ruiz-Gonzalez, F. Vargas-Merino, J. Heredia-larrubia, M. Meco-Gutierrez and F. Perez-Hidalgo, "Application of Slope PWM Strategies to reduce Acoustic Noise radiated by Inverter-Fed Induction Motors," *Industrial Electronics, IEEE Transactions on*, no. 99, p. 1, 2012.
- [13] F. Vargas-Merino, M. Meco-Gutierrez, J. Heredia-Larrubia and A. Ruiz-Gonzalez, "Slope modulation strategy for generated PWM," in *Industrial Electronics, 2008. ISIE 2008. IEEE International Symposium on* , 2008.
- [14] A. Ruiz-Gonzalez, M. Meco-Gutierrez, F. Vargas-Merino, J. Heredia Larrubia and F. Perez-Hidalgo, "New modulation strategy to reduce the acoustic noise and vibrations radiated by PWM -Controlled Induction Machine Drives," in *Electrical Machines, 2008. ICM 2008. 18th International Conference on* , 2008.
- [15] A. Wang and S. Sanders, "On optimal programmed PWM waveforms for DC-DCconverters," in *Power Electronics Specialists Conference, 1992. PESC '92 Record., 23rd Annual IEEE, 1992.*
- [16] I. Takahashi and H. Mochikawa, "Optimum PWM Waveform of an Inverter for Decreasing Acoustic Noise of an Induction Motor," *Industry Applications, IEEE Transactions on* , Volume: IA-22 , Issue: 5 , pp. 828-834, 1986.
- [17] M. Meco-Gutierrez, F. Perez-Hidalgo, F. Vargas-Merino and J. Heredia-Larrubia, "Pulse width modulation technique with harmonic injection and frequency modulated carrier: formulation and application to an induction motor," *Electric Power Applications, IET Volume: 1 , Issue: 5*, pp. 714-726, 2007.
- [18] S. Legowski and A. Trzynadlowski, "Hypersonic MOSFET-based power inverter with random pulse width modulation," in *Industry Applications Society Annual Meeting, 1989., Conference Record of the 1989 IEEE*, 1989.
- [19] T. Nishimura, M. Nakaoka and T. Maruhashi, "Reduction of Vibration and Acoustic Noise in Induction Motor Driven by Three Phase PWM AC Chopper Using Static Induction Transistors," *Power Electronics, IEEE Transactions on* , Volume: 4 , Issue: 3, pp. 313-318, 1989.
- [20] P. Palanivel and S. S. Dash, "Comparative Study of Constant Switching Frequency and Variable Switching Frequency Multicarrier Pulse width Modulation for Three Phase Cascaded Multilevel Inverter," *International Journal of Recent Trends in Engineering*, vol. 2, no. 7, 2009.
- [21] D. Jiang and F. Wang, "Variable Switching Frequency PWM for Three-phase Converter for Loss and EMI Improvement," in *Applied Power Electronics Conference and Exposition (APEC), 2012 Twenty-Seventh Annual IEEE*, 2012.

- [22] J. Pedersen and F. Blaabjerg, "Implementation and test of a digital quasi-random modulated SFAVM PWM in a high performance drive system," in *Industrial Electronics, Control, Instrumentation, and Automation, 1992. Power Electronics and Motion Control., Proceedings of the 1992 International Conference on*, 1992.
- [23] J. A. A. Qahouq, W. Al-Hoor, W. Mikhael, L. Huang and I. Batarseh, "Adaptive Step-Size Digital Controller For Switching Frequency Auto-Tuning," in *IEEE International Symposium on Circuits and Systems, ISCAS'2008*, 2008.
- [24] N. N. Hashemi and R. Lisner, "A New Strategy for Active Control of Acoustic Noise In Converter-Fed Induction Motors," in *AUPEC02*, 2002.
- [25] "Recurrence relation," 30 September 2012. [Online]. Available: http://en.wikipedia.org/wiki/Recurrence_relation. [Accessed 1 December 2012].
- [26] STMicroelectronics, *MEMS digital output motion sensor LIS3DH Datasheet*, 2010.
- [27] *High Performance, Wide Bandwidth Accelerometer ADXL001 Datasheet*, Analog Devices, 2010.
- [28] *LabVIEW*, National Instruments, 2011.
- [29] B. Wilson, "Dealing with Motor Control Dead time Distortion," *Micro Control Journal*, 2001.
- [30] P. Jeong, "The Analysis and Compensation of Dead-Time Effects in PWM Inverters," *IEEE Transactions on Industrial Electronics*, 1991.
- [31] K. Leggate, "Pulse Based Dead Time Compensator for PWM Voltage Inverters," *IEEE Transactions on Industrial Electronics*, 1997.
- [32] Ueda, Sonoda and Takata, "Experimental Results and Their Simplified Analysis on Instability Problems in PWM Inverter Induction Motor Drives," *IEEE Transactions on Industrial Electronics*, 1989.
- [33] D. S. P. N. Muruganatham, "State Space Modeling and Simulation of Sensorless Permanent Magnet BLDC Motor," *International Journal of Engineering Science and Technology*, vol. 2, pp. 5099-5106, 2010.
- [34] "Gradient descent," 7 Novemeber 2012. [Online]. Available: http://en.wikipedia.org/wiki/Gradient_descent. [Accessed 10 November 2012].
- [35] S. Nau and H. Mello, "Acoustic noise in induction motors: causes and solutions," in *Petroleum and Chemical Industry Conference, 2000. Record of Conference Papers. Industry Applications Society 47th Annual*, 2000.

- [36] P. Timar and J. Lai, "Acoustic noise of electromagnetic origin in an ideal frequency-converter-driven induction motor," in *Electric Power Applications, IEE Proceedings, Volume: 141, Issue: 6*, 1994.
- [37] D. Deib and H. Hill, "The Advantages of Harmonic-Distortion Minimization Over Traditional Harmonic-Elimination Techniques," in *Power Electronics Specialists Conference, 1993. PESC '93 Record., 24th Annual IEEE, 1993*.
- [38] A. Wallace, R. Spee and L. Martin, "Current Harmonics and Acoustic Noise in AC Adjustable-Speed Drives," in *Industry Applications Society Annual Meeting, 1988., Conference Record of the 1988 IEEE, 1988*.
- [39] S.-H. Kim and K.-B. Lee, "Elimination of Harmonic Spikes using Second-Order Space Dithered Sigma-Delta Modulation," in *Power Electronics and Applications, 2009. EPE '09. 13th European Conference on, 2009*.
- [40] A. Ruiz-Gonzalez, F. Vargas-Merino, F. Perez-Hidalgo, M. Meco-Gutierrez and J. Heredia-Larrubia, "Low switching PWM strategy to reduce acoustic noise radiated by inverter-fed induction motors," in *Industrial Electronics (ISIE), 2010 IEEE International Symposium on, 2010*.
- [41] . Ruiz-Gonzalez, M. Meco-Gutierrez, F. Perez-Hidalgo, F. Vargas-Merino and J. Heredia-Larrubia, "Reducing Acoustic Noise Radiated by Inverter-Fed Induction Motors Controlled by a New PWM Strategy," *Industrial Electronics, IEEE Transactions on, Volume: 57, Issue: 1, pp. 228-236, 2010*.
- [42] A. Ruiz-González, M. Meco-Gutierrez, F. Vargas-Merino, F. Perez-Hidalgo and J. Ramón Heredia Larrubia, "Shaping the HIPWM-FMTC strategy to reduce acoustic noise radiated by Inverter-Fed Induction Motors," in *Electrical Machines (ICEM), 2010 XIX International Conference on, 2010*.
- [43] W. Lo, C. Chan, Z. Zhu, L. Xu, D. Howe and K. Chau, "Acoustic Noise Radiated by PWM-Controlled Induction Machine Drives," *Industrial Electronics, IEEE Transactions on, Volume: 47, Issue: 4, pp. 880-889, 2000*.