6-9-2016

Modeling Ambient Temperature Affected Photovoltaic Inverter IGBT Lifetimes With Rainflow Analysis

Richard A. Borders

Richard A. Borders
Candidate

Electrical and Computer Engineering
Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Dr. Jane Lehr , Chairperson

Dr. Olga Lavrova

Dr. Edward Graham

MODELING AMBIENT TEMPERATURE AFFECTED PHOTOVOLTAIC INVERTER IGBT LIFETIMES WITH RAINFLOW ANALYSIS

by

RICHARD A. BORDERS

PREVIOUS DEGREES
B.S. ELECTRICAL ENGINEERING

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico
Albuquerque, New Mexico

May, 2016

DEDICATION

I dedicate this thesis to my loving wife, Megan Borders. Without her endless support and understanding, it is unlikely that I would have been able to complete the task. With a new family member just months away, we can now move on to the next stage of our lives together with one less commitment, though many more are sure to come.

ACKNOWLEDGMENTS

I would like to acknowledge Olga Lavrova, my advisor for this project, for guiding me in the right direction regarding a project subject and helping me along when I was stuck in my simulations. I would also like to thank Tracy Peterson, my manager at work, for allowing me the flexibility to complete this project as well as all my classes around a full workweek. Finally, I would like to thank Marcus Craig, my senior manager at work, for providing me with the motivation to complete this project in a timely fashion.

MODELING AMBIENT TEMPERATURE AFFECTED PHOTOVOLTAIC INVERTER IGBT LIFETIMES WITH RAINFLOW ANALYSIS

by

Richard A. Borders

B.S. Electrical Engineering, University of Southern Maine, 2011

M.S. Electrical Engineering, University of New Mexico, 2016

ABSTRACT

This manuscript details the modeling performed towards estimating lifetimes of IGBTs in PV inverters. Specifically, ambient temperature, solar irradiance, and load demand for a duration of a year were considered in modeling for a chosen integrated gate bipolar transistor (IGBT) from International Rectifier. Two cases were considered in the modeling, yielding lifetimes of 29.1 years and 4.49 years. Mean time to failures (MTTF) were calculated for an H-bridge topology inverter with four IGBTs for both cases as 7.27 and 1.22 years, respectively.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 – MOTIVATION

This manuscript details the work I performed in order to model ambient temperature affected photovoltaic inverter IGBT (integrated gate bipolar transistor) lifetimes, specifically with the implementation of rainflow analysis. The motivation for this work is varied, but can ultimately be reduced to economics.

Photovoltaic (PV) energy is a rapidly growing industry. As testament to that statement is the fact that 1GW of PV was installed in 2011 in the United States, up from just 50MW in 2004 [1]. Even though PV installations are on the rise as an overall percent of energy generation, PV lags behind traditional generation technologies such as coal and natural gas. This is in large part because of still-elevated costs relative to the older technologies.

Levelized cost of energy (LCOE) is a metric that can be used to compare different energy technologies from a financial standpoint that takes into account lifetime energy production as well as lifetime costs. LCOE can include any number of costs in its calculation to include financing, insurance, maintenance, etc. Additionally, LCOE can essentially be used to determine at what price energy needs to be sold in order for a given technology to be profitable.

Part of the costs that go into LCOE calculations for PV systems have to do with their inverters, which are a critical assembly, especially as related to system failures. Digging down further, IGBTs are the second most unreliable components in inverter designs, second only to capacitors [2]. IGBTs, then, are an important component to hone in on because system degradation has been shown to significantly contribute to variability in LCOE calculations [1]. For IGBTs in inverters, the most commonly observed failure modes surround temperature stresses to include temperature swings as well as the mean temperature of those swings [3].

The fact that IGBTs are sensitive to temperature is unfortunate, as they typically have rigorous demands that require high degrees of reliability for a variety of applications [4]. Aside from their aforementioned application in PV inverters, they are commonly used in inverters for wind turbines [5] and traction applications.

To quantify a component’s life is to quantify its durability [6]. That, essentially, is the goal of the work presented here.

CHAPTER 2 – INTRODUCTION

2.1 IGBT Construction and Applications.

IGBT construction is somewhat similar to that of a metal oxide semiconductor field effect transistor (MOSFET), but they have an additional p+ layer. IGBTs can be thought of as something of a cross between a MOSFET and a bipolar junction transistor (BJT) [7]. Figure 1 shows the additional p+ layer that IGBTs possess as compared to the makeup of a MOSFET.

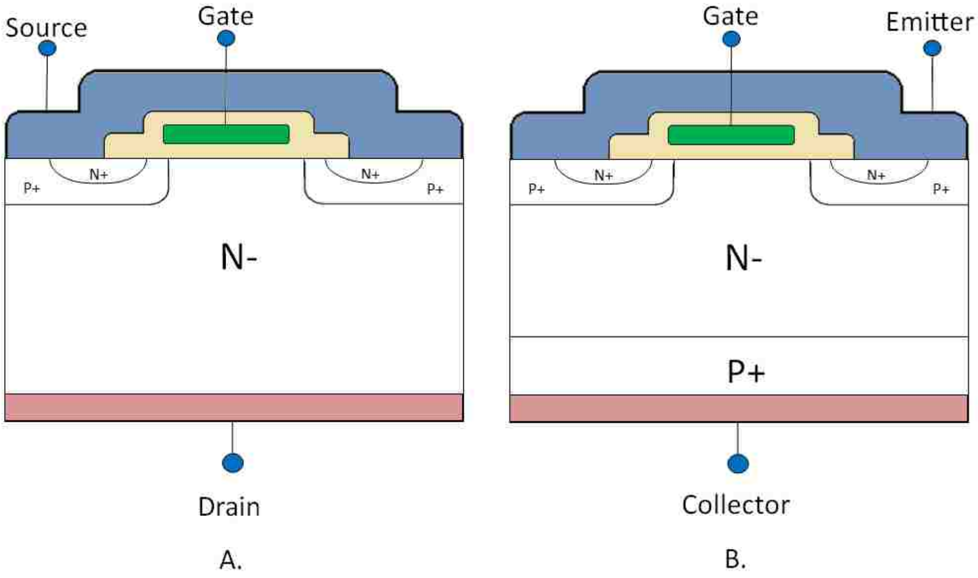


Figure 1. A. MOSFET cross section, B. IGBT cross section

IGBTs are used in a variety of applications to include variable speed drives [4], wind turbine applications [8], and PV inverters, among others. Some of the advantages of IGBTs are that they work very well in applications requiring switching speeds up to 20kHz, they have low on-resistance, have a simple gate drive, and possess high blocking voltage ratings [9]. For applications requiring high current carrying capabilities, IGBTs meet the need as well; typically several sub components are placed in parallel within a single IGBT package. In converter topologies, IGBTs are typically expected to last on the order of 10 years or 80,000 service hours. For certain applications, like wind turbines, they are expected to remain functional for twice as long – approximately 20 years [8]; for traction applications, only one device failure in 10^{11} component-hours is allowed [4].

2.2 IGBT Reliability.

Early efforts towards determining the reliability of IGBTs focused specifically on lifting bond wires and solder joint fatigue. The majority of problems that arise regarding IGBT reliability stem from thermal problems specifically due to coefficient of thermal expansion (CTE) mismatches between different components of the device. For example, the CTE of silicon (Si) is $3x10^{-6}$/K, $4.5x10^{-6}$/K for ceramic, $17x10^{-6}$/K for copper (Cu), and $23x10^{-6}$/K for aluminum (Al).

Early IGBTs were most prone to failures stemming from bond wire liftoff, but recent advancements have allowed for solder fatigue to compete as a predominant failure mode [10]. More specifically, IGBT weaknesses include the bond wire to Si chip interface, the Si chip to DCB (direct copper bond, which is a direct Cu-ceramic bond) solder interface, and the DCB to base plate solder interface. CTE mismatch between the DCB and base plate is of most concern with respect to solder joint fatigue. Deterioration of this solder joint leads to reduced heat dissipation, which in turn leads to increased junction temperatures (T_j), resulting in higher failure rates.

The reliability of the wire bond, is problematic because the emitter-gate has an Al wire bonded to the Cu contact on the DCB [4]. Specifically, bond wire failures occur as a result of crack propagation that is caused by thermo-mechanical stresses as a result of CTE mismatches [9]. Regarding solder layers, the solder between the Cu-DCB also has mechanical stresses imparted on it from CTE mismatches. Here, when the DCB and solder delaminate, there is a resultant increase in the thermal resistance, which consequently leads to thermal runaway. Press-pack IGBT technology has been developed that eliminates liftoff and solder joint fatigue, but this technology introduces other failure modes.

It is important to note that internal or external heating can be the cause of concern for IGBTs. For example, IGBT power cycling leads to increased values for change of temperature (ΔT), which in turn leads to mechanical stresses, which ultimately lead to shorter failure times. This ΔT would be an example of an internal change; seasonal and daily changes as well as other surrounding-equipment caused changes would be external.

IGBT technology development has always pushed for greater power density in order to reduce cost. Unfortunately, as power density increases, a concomitant increase in current occurs. This, in turn, leads to an increase in IGBT junction temperature, leading to an increase in stress, and ultimately a decrease in the number of cycles to failure for the component [8], [11].

Of the failure modes discussed above, it is important to make some distinctions. Qualitatively, several different types of failures exist: hard, soft, and performance. A hard failure, on one hand, would be if the IGBT ceased to function altogether. A soft failure, for example, would be if the IGBT ceased to function only at high temperatures. This might be possible if the bond wire was lifting off at high temperatures, but then returning to place at lower temperatures. Finally, a performance failure could be conduction losses permanently increasing because of the degradation of solder joints, though the device might still function in this reduced capacity.

For reliability testing, acceleration factors can be determined by testing various parameters at extreme values to include temperature, humidity, change in temperature, mechanical vibration, and mechanical shock. Testing detailed in [9] yielded a descriptive model for IGBT failures as a function of mean junction temperature, T_{jm}, as well as the change in junction temperature, ΔT_j. Note that this model has a Coffin-Manson term in addition to an Arrhenius term:

$$N_f = A \times \Delta \mathrm{T}_j{}^{\alpha} \times e^{Q/_{R \times T_{jm}}} \tag{1}$$

In Eq. 1, N_f represents the number of cycles to failure; A was empirically determined to be 640, $\alpha = -5$, $Q = 7.8 \times 10^4$ J/mol (~0.8 eV), and R = 8.314 J/(mol x K) – the ideal gas constant.

2.3 Thermal Considerations.

Industry has adopted the idea of transient thermal response. Specifically, this is the thermal response to a step input of power. Transient thermal impedance (more commonly known as R_{th} or Z_{th}) vs. pulse-width curves are commonly reported by manufacturers. They can be generated either empirically or with models [12].

Semiconductor thermal behavior can be described by several different models. The continued fraction model is one of them; it is also known as the Cauer, T, or Ladder model. This model actually represents the layer sequence in the semiconductor. The partial fraction model is another; it is also known as the Foster or Pi model [13]. For the partial fraction model, the thermal impedance equation is:

$$Z_{th}(t) = \sum_{i=1}^{n} r_i \, x \, (1 - e^{-t/\tau_i}) \tag{2}$$

In Eq. 2, r_i and τ_i are empirically determined constants.

IGBT power losses are dissipated through junction-to-case, case-to-heatsink, and heatsink-to-ambient thermal impedances. The temperature at the IGBT's junction can be determined by using Eqs. 3 and 4 shown next:

$$T_j(t) = P_{total}(t) \times Z_{th,jc}(t) + T_c(t) \tag{3}$$

$$T_c(t) = T_a(t) + [P_{IGBT}(t) + P_{diode}(t)] \times [Z_{th,ch}(t) + Z_{th,ha}(t)] \tag{4}$$

In Eqs. 3 and 4, T_j is the junction temperature, T_c is the case temperature, T_a is the ambient temperature, P_{total} is the total power dissipated by the component (IGBT + diode, should it include one in the package), P_{IGBT} is the power dissipated by the IGBT, P_{diode} is the power dissipated by the diode, $Z_{th,jc}$ is the thermal impedance from junction to case, $Z_{th,ch}$ is the thermal impedance from case to heatsink, and $Z_{th,ha}$ is the thermal impedance from heatsink to ambient.

Occasionally, device manufacturers do not report Z_{th}, but instead report R_{th}. Typically, R_{th} is simply the maximum value of Z_{th}. Conveniently, the terms are interchangeable in Eqs. 3 and 4. Sometimes a value for $Z_{th,ja}$, thermal impedance from junction to ambient, can be found; this could be utilized in Eq. 3 by replacing $Z_{th,jc}$ and dropping the case temperature term.

2.4 Rainflow Analysis.

Cycle counting methods are used for life prediction when the load is variable, for example, a temperature load. Examples of cycle counting methods include peak counting, level counting, Hayes method, racetrack range counting, and rainflow analysis [2].

Rainflow analysis determines closed hysteresis loops for changing amplitude loads [14]. Rainflow can also account for changing temperatures during a cycle, although this complicates the analysis. The algorithm for rainflow counting is thoroughly described in [15]. Essentially, rainflow allows for a fatigue calculation to be performed over a long-term stress distribution, especially in cases where the stress distribution is not evidently periodic within a narrow time band.

For material behaviors that are non-dependent on rate and when constant mean temperature is held, the rainflow analysis only requires the reversal points in the time history. If a non-constant mean temperature is used, then the mean temperature for each individual cycle is required for the analysis. When temperature variations are considered, more accurate life estimates can be yielded [12].

Annual temperature profiles lend themselves to this kind of analysis very well, as there is not a significant degree of periodicity on the sub-year scale. Using a rainflow method on an annual time history would yield two important strings of data as related to IGBTs: mean junction temperature and change in junction temperature. Knowledge of the load, cooling conditions, and the IGBT parameters are all important for precise calculations of T_{jm} and ΔT_j, which ultimately leads to an accurate lifetime calculation [8].

Once the rainflow analysis is complete, the Palmgren-Miner rule, also known simply as Miner's rule, can be used to calculate the damage to the component. This rule is described by equation 5:

$$\sum_{i=1}^{k} {}^{n_i}/_{N_i} = C \qquad (5)$$

In Eq. 5, n_i is the number of cycles observed at a given stress range, N_i is the number of cycles that would cause failure at a given stress range, and $C = 1$ is typically assumed to be the failure point for design purposes.

CHAPTER 3 – PRELIMINARY LIFETIME SIMULATIONS

3.1 Lifetime Estimation Accounting Only for Ambient Temperature.

I obtained annual temperature data at hourly intervals from the National Renewable Energy Laboratory (NREL) National Solar Radiation Database (NSRDB) [16]. Being a student at the University of New Mexico, I chose data from the Albuquerque International Sunport for the 2014 calendar year.

The temperature profile for Albuquerque is shown in Fig. 2, plotted with Matlab. The data starts at 12 a.m. on January 1st, 2014. The series has 8760 points, which is the number of hours in a year (leap-years aside). Note the expected overall trend of the data, starting cold in the winter, warming up throughout the year, and then cooling back down towards then end as the next winter approaches.

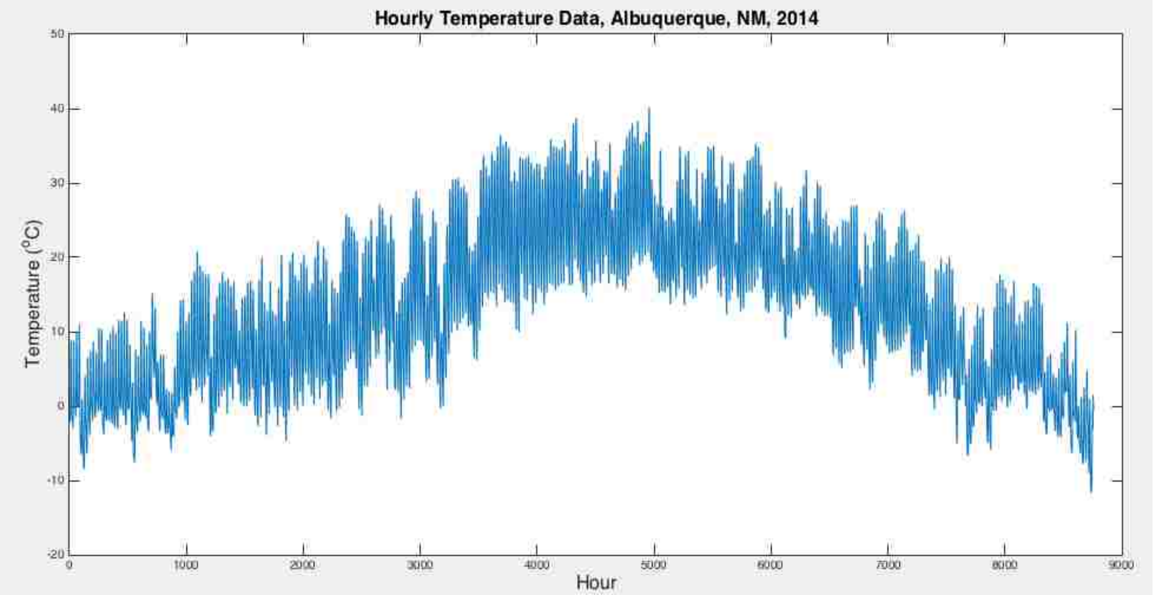


Figure 2. Hourly Temperature in Albuquerque for 2014.

My first goal was to determine what the lifetime of the IGBT in the PV inverter would be making some gross assumptions, specifically that the change in junction temperature of the device, ΔT_j, was exactly the ambient temperature profile and the mean junction temperature, T_{jm}, was 100°C. Using a T_{jm} of 100°C, I used Eq. 1 to generate the plot shown in Fig. 3.

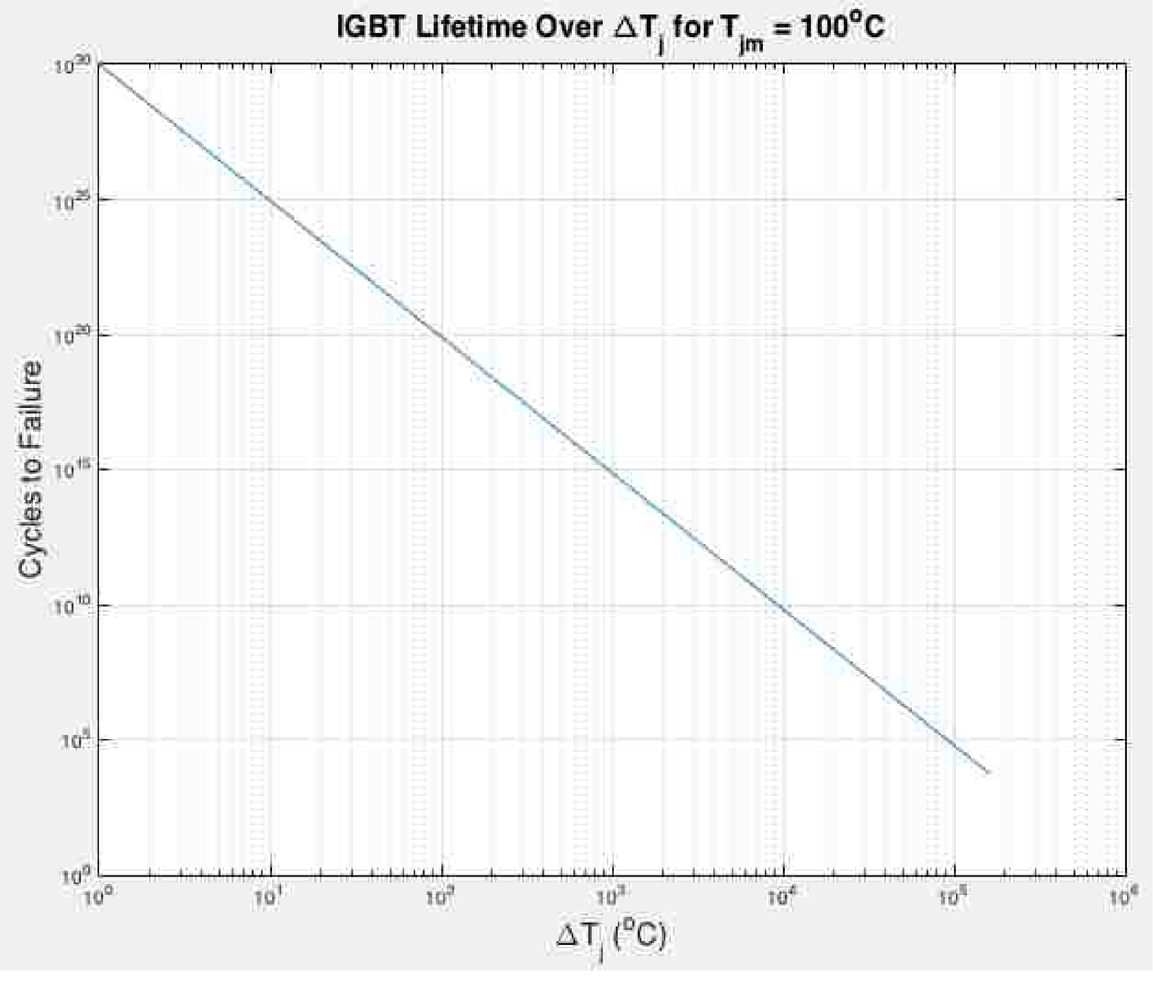


Figure 3. IGBT Lifetime Over ΔT_j for a T_{jm} of 100°C.

Next, I had to run a rainflow analysis on the temperature data in order to correlate it to the number of cycles to failure. I used the Rainflow Counting Algorithm by Nieslony and available on MathWorks' File Exchange to perform the analysis [17]. It is important to note that the input to The Rainflow Counting Algorithm has to be the turning points of the time history, not the entire time history itself. Once input, the algorithm has the capability to provide some interesting output. For example, Fig. 4 shows the cycle extraction from the input signal.

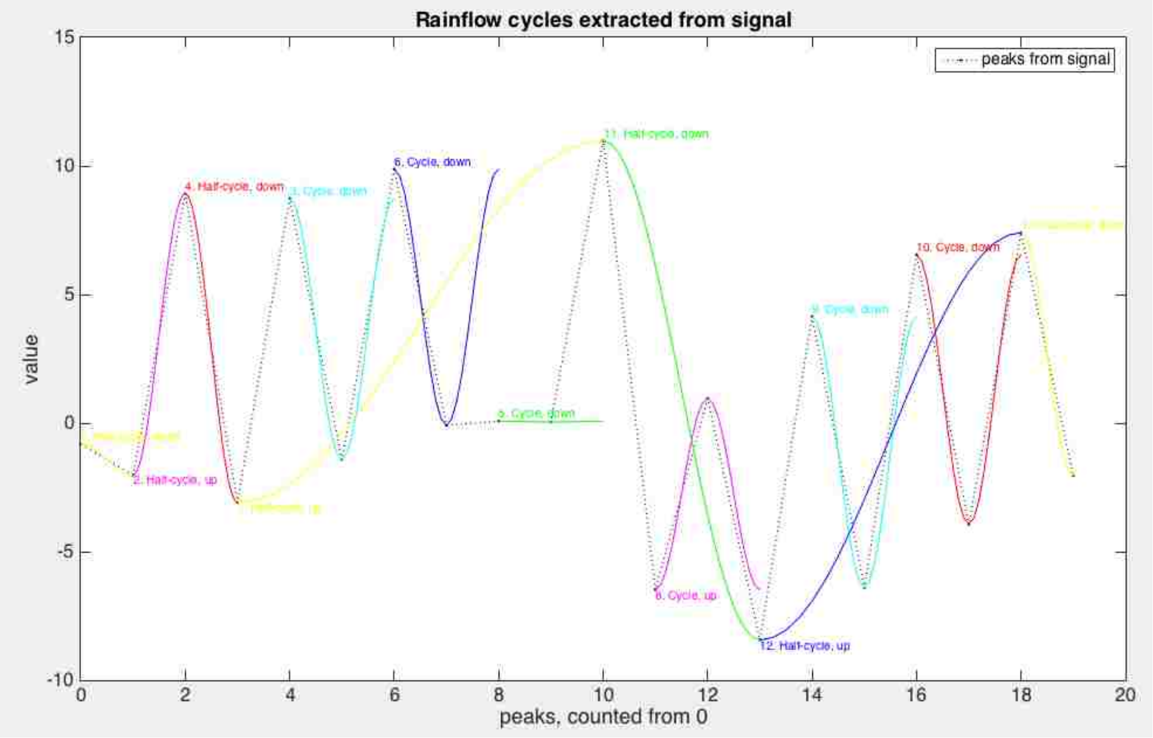


Figure 4. Cycle Extraction from Turning Points of Temperature Data.

More importantly, the algorithm outputs the raw data for the cycle amplitudes, which represent ΔT_j and are required for lifetime analysis. Using Eq. 5 and the output shown in Fig. 3, I calculated that the lifetime of the device would be 8.797x10^4 years. Clearly, this first lifetime estimate is far from realistic.

3.2 Lifetime Estimation Accounting for Attenuated Ambient Temperatures.

The next step I decided to take in order to approach a more realistic thermal transfer model was to attenuate the temperature signal so that ΔT_j values would not be so high. For this step, I arbitrarily attenuated the temperature signal with a lowpass filter of order 8 and a cutoff of 0.08. This yielded a new temperature profile, shown in Fig. 5.

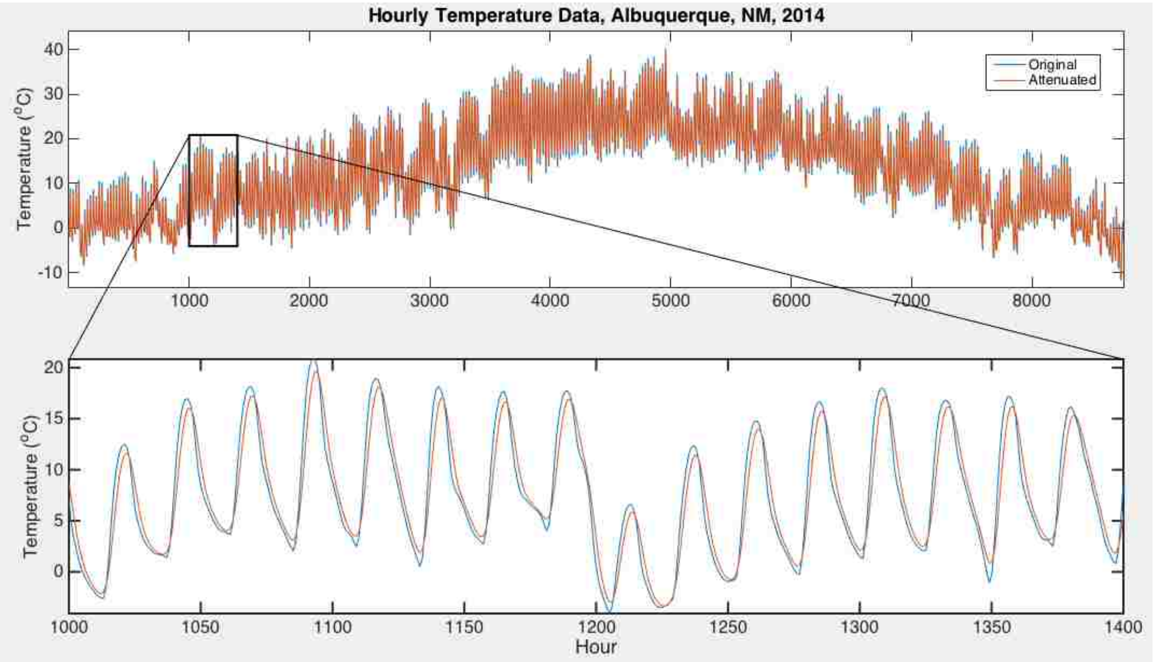


Figure 5. Attenuated Temperature Profile with Filter Order 8.

Figure 5 shows that the attenuation is not severe, but noticeable. Running the rainflow analysis on this new profile and Miner's damage rule, I calculated an increased lifetime of $1.104\text{x}10^5$ years. I continued to step up the attenuation by increasing the filter order by a factor of two all the way up to an order of 512. Figure 6 shows a 150-point section of the family of attenuated temperature profiles.

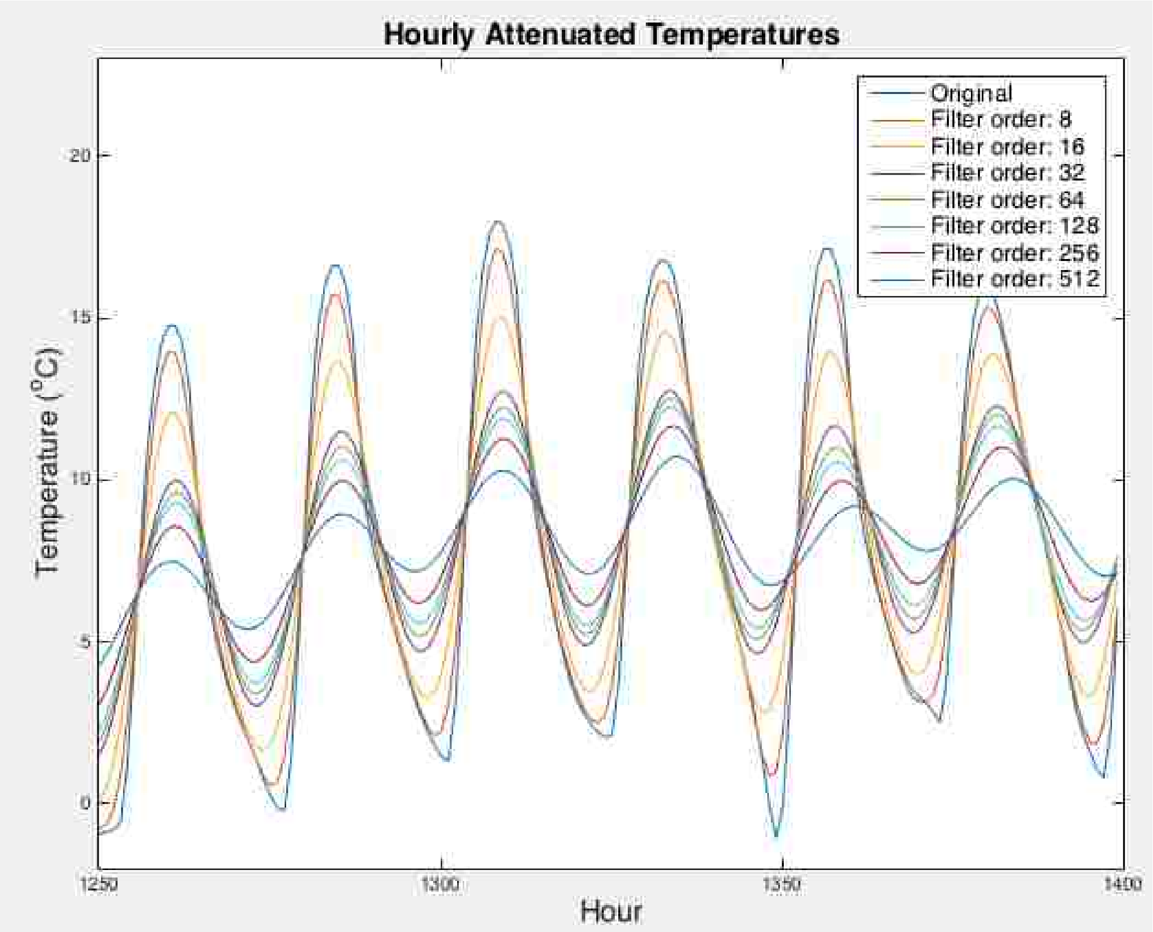


Figure 6. Attenuated Temperature Curves.

It is clear that by the time the filter order of 512 is reached, the attenuation becomes quite significant. Running rainflow analysis and the damage calculation, I generated the lifetimes at all filter orders. This is shown in Table 1.

Table 1. IGBT Lifetimes for Attenuated Temperature Profiles, $T_{jm} = 100^{o}C$.

Filter Order	Lifetime (years)
[Original]	87,966
8	110,453
16	166,139
32	277,696
64	290,273
128	308,577
256	345,456
512	426,424

These values are presented in graphical form in Fig. 7:

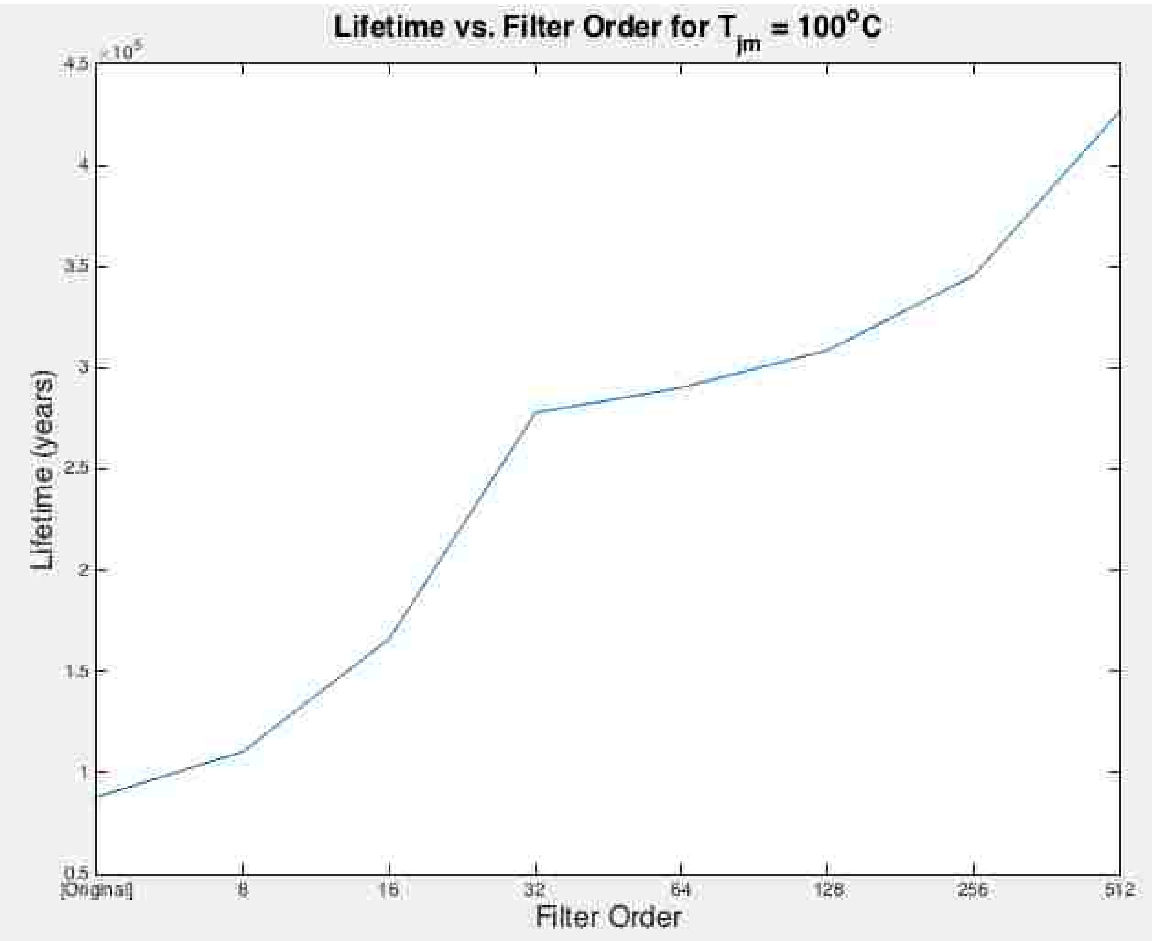


Figure 7. IGBT Lifetimes for Attenuated Temperature Profiles, T_{jm} = 100°C.

Note from Table 1 and Fig. 7 that the lifetimes I've calculated are wildly long. This to some degree makes intuitive sense – if the device were not powered and were just sitting at ambient temperature fluctuations, it would be expected that the device would last a very long time. Of course, it is unlikely that they would last that long; instead some other failure mode would likely take over, perhaps such as the deterioration of the materials.

3.3 Lifetime Calculations with Attenuated Ambient Temperatures and Varied T_{jm}.

The next step would be to arbitrarily vary T_{jm} to different values. I decided to start by additionally considering T_{jm} at 60°C and 80°C; these are plotted along with the original T_{jm} of 100°C as cycles to failure vs. ΔT_j in Fig. 8. It is important to note that with decreased values for T_{jm}, the lifetime at any given ΔT_j increases – this is in agreement with Eq. 1.

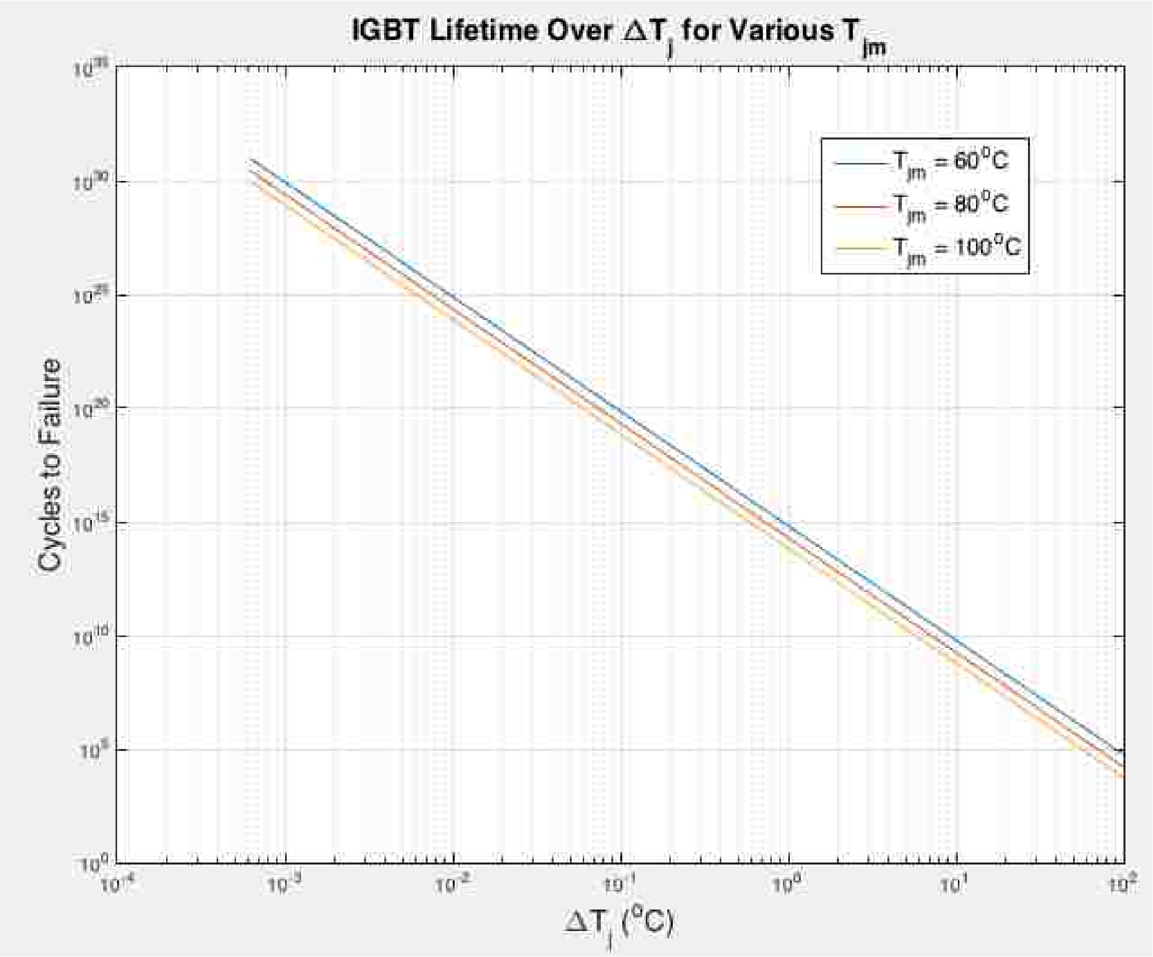


Figure 8. IGBT Lifetime Over ΔT_j for Various T_{jm}.

Using these new failure curves, I calculated failure times for the original temperature profile as well as the attenuated ones shown in Fig. 6. The calculated lifetimes are tabulated in Table 2.

Table 2. IGBT Lifetimes for Attenuated Temperature Profiles at Various T_{jm}.

Filter Order	Lifetime (years)		
	T_{jm} = 60C	T_{jm} = 80C	T_{jm} = 100C
[Original]	883,710	261,173	87,966
8	1,109,605	327,935	110,453
16	1,669,031	493,269	166,139
32	2,789,718	824,479	277,696
64	2,916,070	861,822	290,273
128	3,099,950	916,166	308,577
256	3,470,437	1,025,661	345,456
512	4,283,837	1,266,055	426,424

These values are presented in graphical form in Fig. 9:

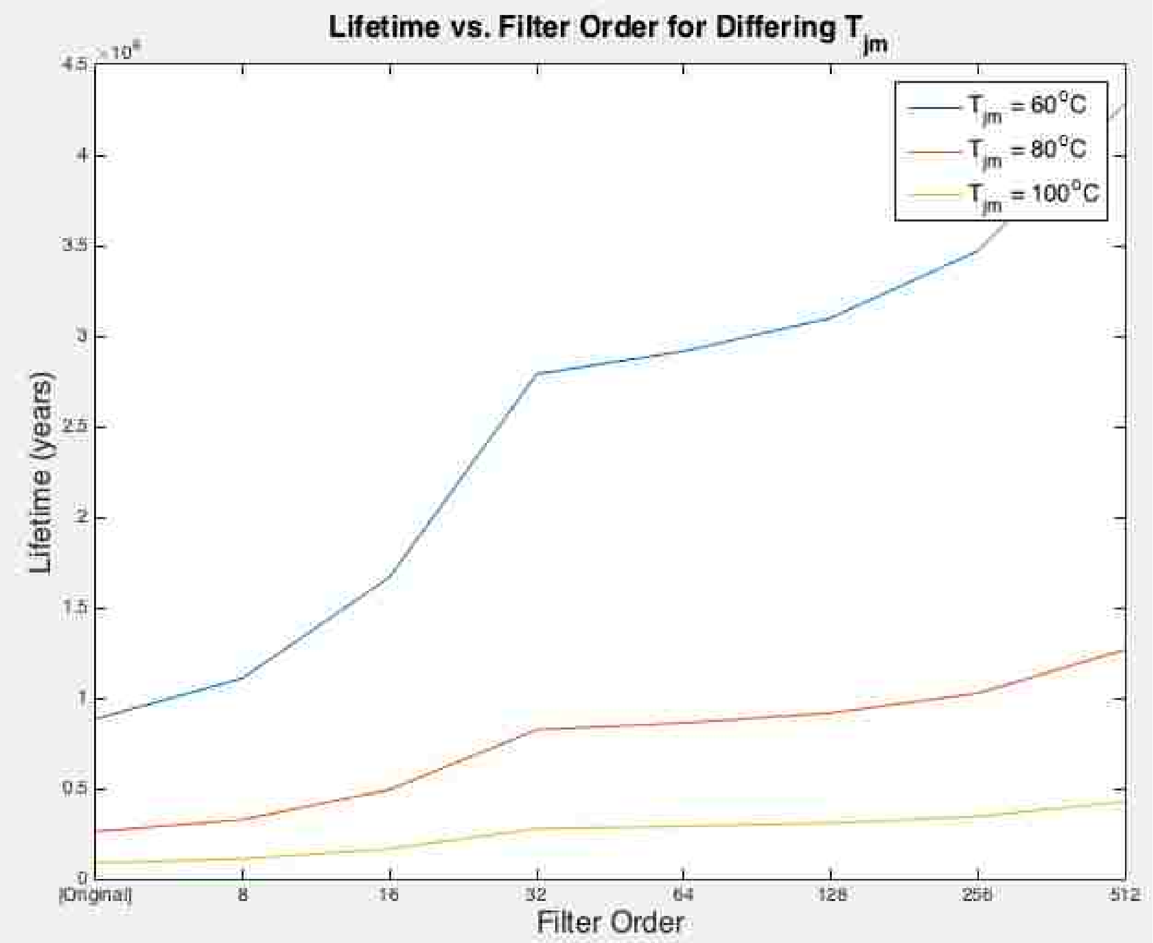


Figure 9. IGBT Lifetimes for Attenuated Temperature Profiles at Various T_{jm}.

From Table 2 and Fig. 9, it is clear that the lifetime of the IGBT increases with decreasing T_{jm} as well as increasing filter order.

CHAPTER 4 – LIFETIMES WITH VARIABLE T_{jm}

4.1 Lifetimes with Attenuated Ambient Temperatures and Variable Calculated T_{jm}

The next step required me to account for a changing T_{jm} for each cycle in the rainflow analysis. Conveniently, another output from The Rainflow Counting Algorithm [17] is the cycle mean value, which here represents T_{jm}.

Using this data, it is important to note here that Matlab was unable to immediately calculate a lifetime based on calculated T_{jm}'s from the rainflow analysis, as some values were too low. Very low T_{jm}'s cause the output from the calculated cycles to failure curves to quickly approach infinity. This is specifically because of the sensitivity of Eq. 1 to T_{jm}, as it is in the denominator of the exponent for the Arrhenius term. For example, Matlab calculates that the lifetime for a ΔT_j of 10°C and a T_{jm} of 10.108°C is infinity, whereas increasing T_{jm} to 10.109°C results in a lifetime of 1.6882×10^{308} years. This occurs because Matlab's largest positive floating point number is based on the largest finite floating-point number in IEEE double precision: 1.7977×10^{308}. The minimum value of ΔT_J from the rainflow analysis in my case is 6.1839×10^{4}, which means the minimum allowable value for T_{jm} is approximately 10.9°C. As such, I arbitrarily shifted the calculated T_{jm} values up to a maximum of 100°C, yielding a minimum of approximately 65°C. This was assumed to be a good place to start regarding realistic junction temperatures for the IGBT.

Figure 10 shows the original mean junction temperatures that are output from the rainflow analysis as well as the adjusted values I used for lifetime calculations for the filter order 8 case, as an example.

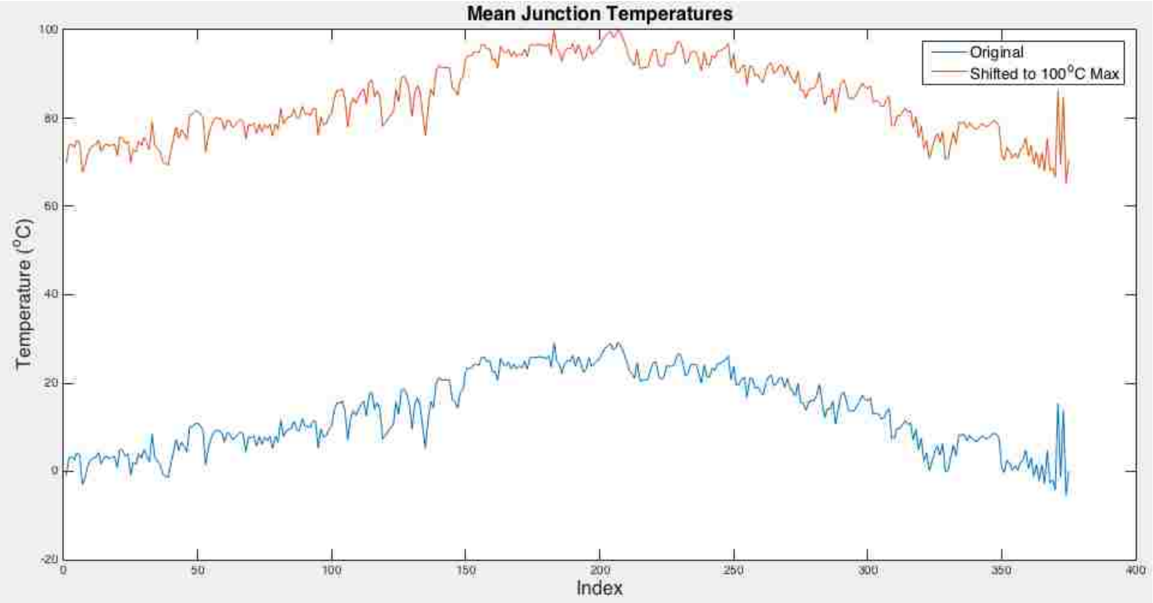


Figure 10. Original and Shifted T_{jm} for Filter Order 8.

Another problem encountered was for very low ΔT_j. In order to account for all ΔT_j, my calculations would overflow because of a request to create a matrix for the number of cycles to failure that was too large for Matlab to handle (e.g. 19GB). As such, values of ΔT_j that were below 0.001°C were rounded up to 0.001°C. Looking at Figs. 3 and 8, it is clear that very low values of ΔT_j do not significantly impact the lifetime of the IGBT.

With overflow problems addressed, lifetimes for the different degrees of attenuation on the original temperature profile are shown in Table 3.

Table 3. IGBT Lifetimes for Attenuated Temperature Profiles for Varying T_{jm}.

Filter Order	Lifetime (years)
[Original]	185,140
8	236,123
16	375,146
32	648,668
64	867,113
128	944,919
256	1,210,834
512	1,589,948

These values are presented in graphical form in Fig. 11:

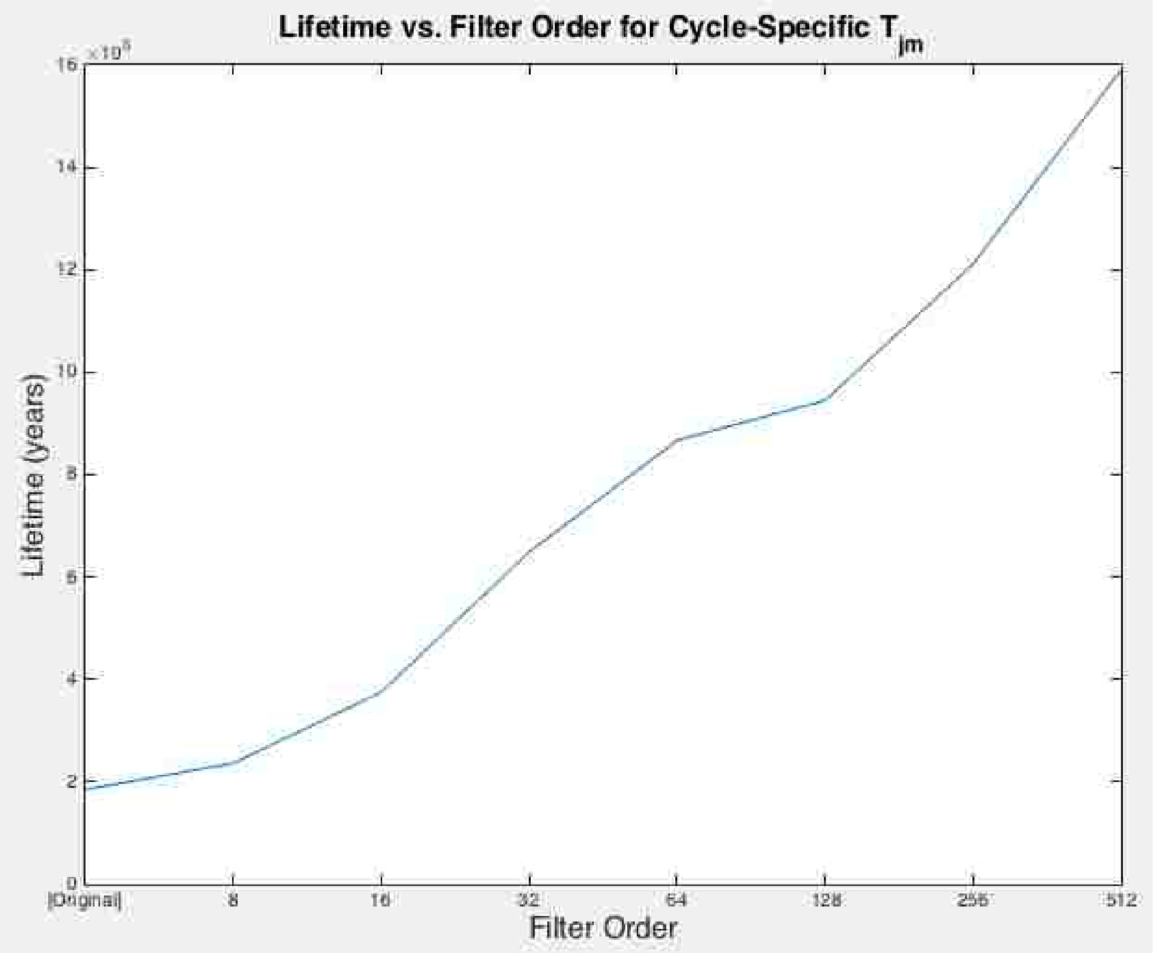


Figure 11. IGBT Lifetimes for Attenuated Temperature Profiles for Varying T_{jm}.

Note in Table 3 and Fig. 11 that the lifetimes that are shown lie between the lifetimes that are reported in Table 2 and Figure 9. This makes intuitive sense as the mean junction temperatures are generally between 60°C and 100°C, which are the bounds in Table 2 and Fig. 9.

All the above was based solely on ambient temperature variations. The next step would be to consider calculated junction temperatures.

CHAPTER 5 – LIFETIMES WITH POWER OUTPUT CONSIDERATIONS

5.1 Case Descriptions.

In order to calculate the junction temperature with Eqs. 3 and 4, I evaluated two cases. In the first case, the inverter is designed to output max power when the irradiance is highest *and* the load demand is highest. That means there is no output when there is no demand, nor when there is no irradiance. In the second case, the inverter is designed to support max load when the irradiance is above 50% of its maximum value. Case two could be realized, for example, by connecting a roughly two times larger PV array to the same inverter. In both cases, maximum power point tracking (MPPT) is assumed as well as that power output is linear with irradiance. Case 1 is obviously a very poor design with zero margin, whereas case two is slightly more realistic. Additionally, I will also assume that power dissipated in the inverter's IGBTs is linearly proportional to the output of the inverter itself, i.e. when max power is output by the inverter, max power is dissipated in the IGBT.

The global horizontal irradiance (GHI) for the Albuquerque International Sunport was downloaded, as with the temperature data, from NREL's NSRDB. This is shown in Fig. 12.

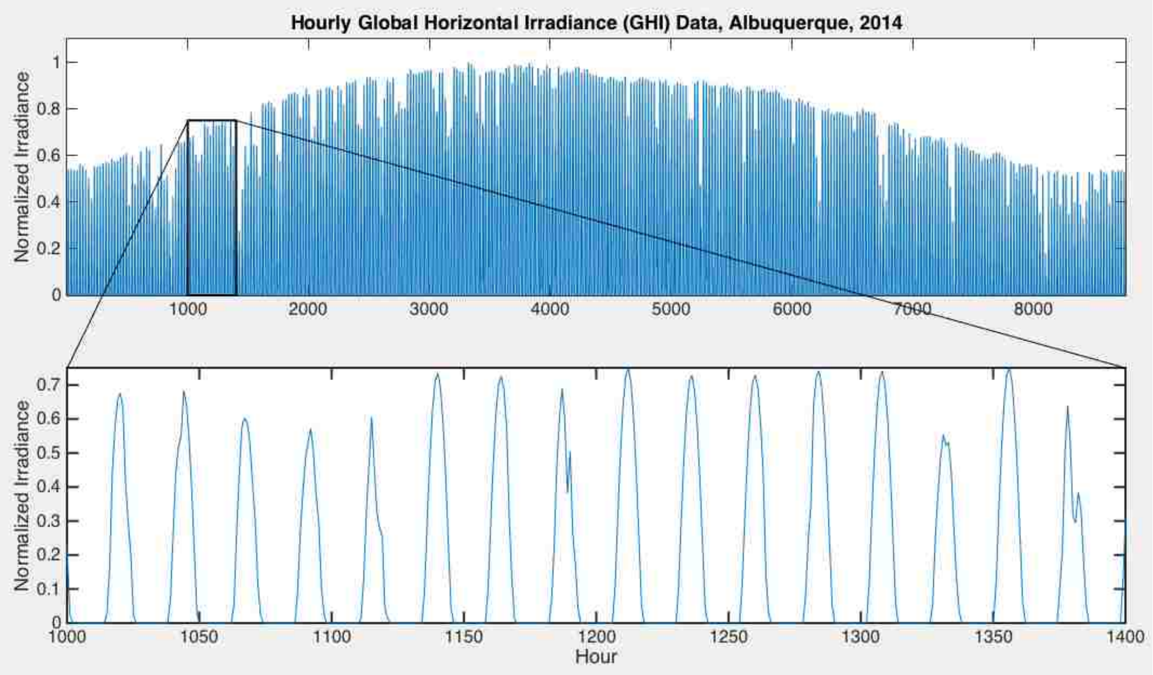


Figure 12. Normalized Hourly Global Horizontal Irradiance in Albuquerque for 2014.

Note in Fig. 12 that there is the characteristic irradiance curve at the daily time scale. Specifically, looking at the lower, zoomed-in plot, it is clear that the irradiance is zero, and then gradually increases to some maximum value, and decreases back to zero again before repeating. This is of course in sync with the daily rise and fall of the sun in the sky. There are occasionally drops in the irradiance in the middle of the day, likely corresponding to cloud cover. Also, note that the irradiance is normalized.

Next, I found an annual load profile. As I was unable to locate one for anywhere in Albuquerque, I used one in its place from New York City [18]. Figure 13 shows this annual load profile.

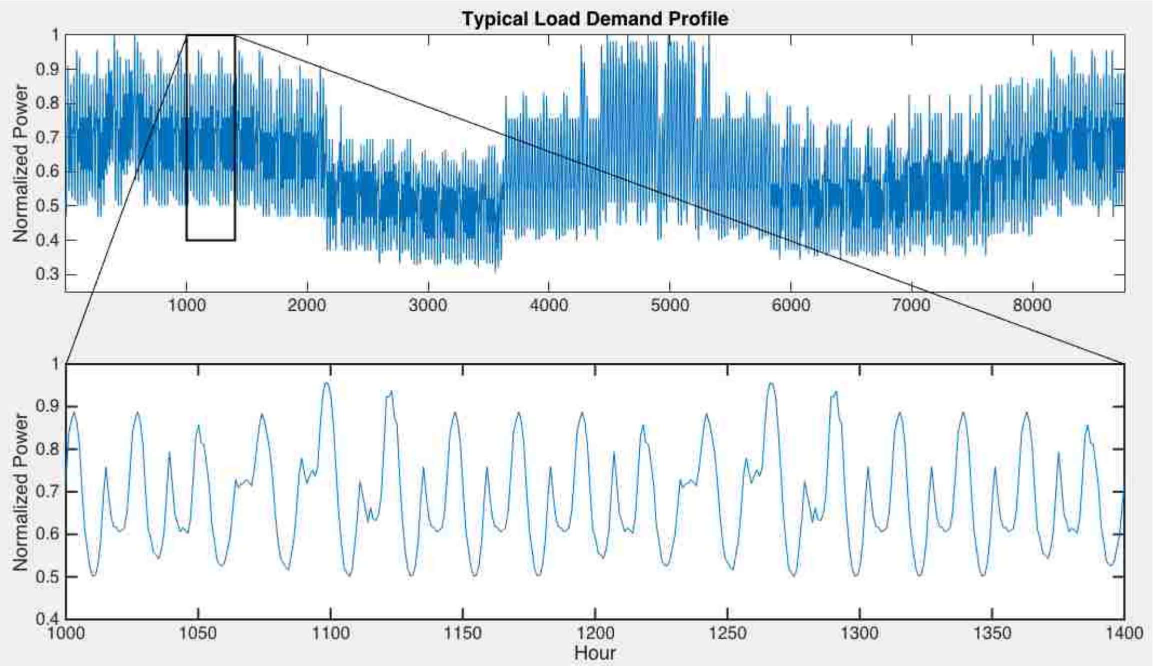


Figure 13. Normalized Hourly Load Profile.

As with the GHI data in Fig. 12, the load data in Fig. 13 is normalized. The familiar daily load curve shape is visible in the zoomed-in portion of Fig. 13. For example, one can see the load rise from approximately hour 1150 to some intermediate value – this corresponds to the morning load where people are waking up and getting out the door. The load subsequently drops, which corresponds to people being at work or school. Then the load peaks for the day when people arrive back home in the evening and perhaps turn on the TV and cook dinner. Finally the load drops to its local minimum again as people go to sleep for the night.

5.2 Case 1 Lifetime Estimation.

For the first case to be evaluated, the product of the normalized irradiance and normalized demand, then, gives normalized power output. This is shown in Fig. 14.

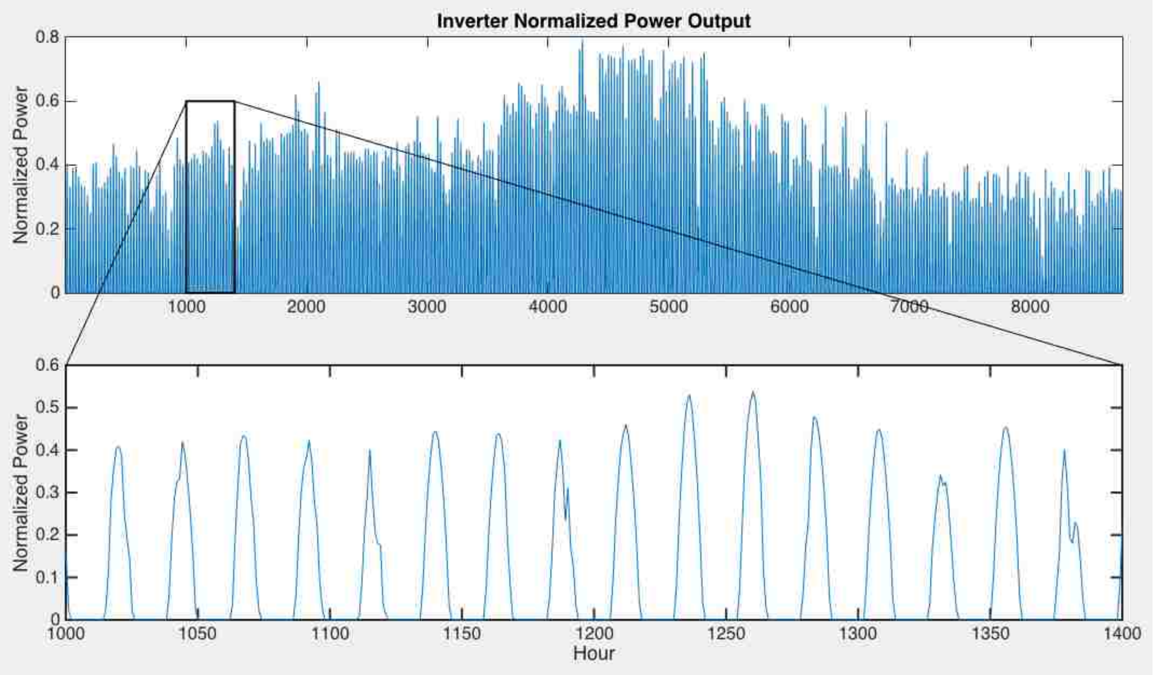


Figure 14. Normalized Hourly Power Output Profile, Case 1.

I chose to proceed with my modeling using International Rectifier's IRGP4063DPbF IGBT. Looking at the datasheet for this device (Appendix I) one can see that P_{tot}, or the power dissipation capability, is not a constant value, but rather a function of the IGBT case temperature. Specifically, P_{tot} is linear between case temperatures of 25°C and 100°C, dropping from 330W to 170W. Linearity is maintained up to 175°C, where Ptot = 0. At T_c less than 25°C, P_{tot} is constant at 330W. The P_{tot} vs. T_c plot shown in the datasheet is recreated here in Fig. 15. Its recreation was necessary in order to accurately calculate the power dissipation at any given point in the yearly timeframe for the IGBT.

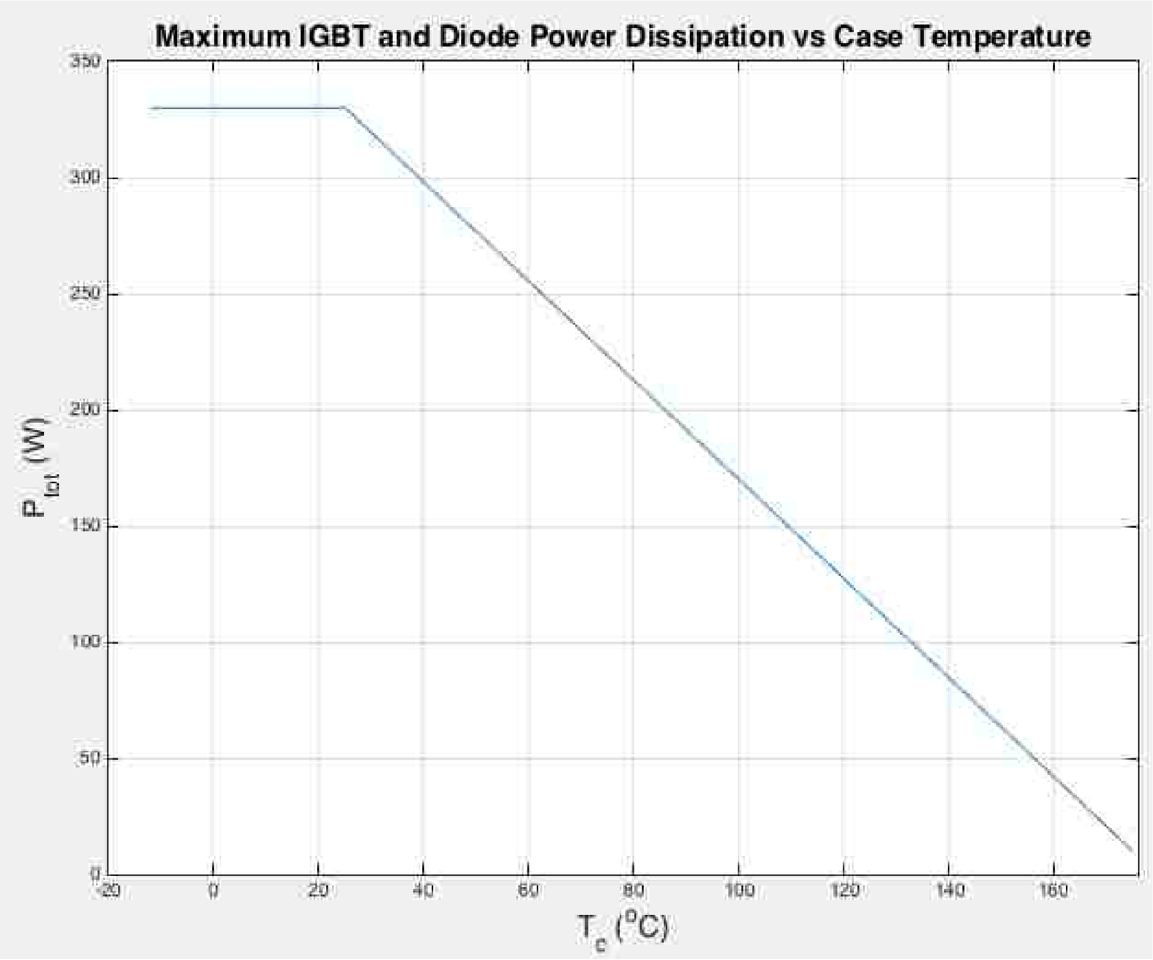


Figure 15. Maximum IGBT Power Dissipation as a Function of T_c.

Note that the line in Fig. 15 is the *maximum* power dissipation possible. My assumption, mentioned earlier, is that the IGBT will dissipate the maximum power when the power output, shown in Fig. 14, is at its peak. I assume the relationship to be linear, so if power output is 50%, then power dissipated is 50% of the maximum value for a given case temperature. This loss would specifically be a conduction loss. I also account for switching losses by adding that to the conduction loss up to the maximum allowable value for dissipation. The final item to consider is that since power dissipation relies on the case temperature, I would need a time t = -1 case temperature value in order to determine the power dissipation value at time t = 0. In lieu of finding the temperature data for the previous year, 2013, and picking the last point from that, I simply set the t = -1 case temperature to the first temperature value of the annual ambient temperature. This would be representative of the inverter starting up at the first hour of the year, having been powered down and at equilibrium with ambient prior.

I recall here for quick reference Eqs. 4 and 5:

$$T_j(t) = P_{total}(t) \times Z_{th,jc}(t) + T_c(t) \tag{3}$$

$$T_c(t) = T_a(t) + [P_{IGBT}(t) + P_{diode}(t)] \times [Z_{th,ch}(t) + Z_{th,ha}(t)] \tag{4}$$

Now that power dissipation has been determined, the next values to determine are the thermal impedances. Recalling from section 2.3 that R_{th} and Z_{th} can be interchanged in Eqs. 3 and 4, I used values from the IGBT datasheet as well as [19] for the junction-to-case, case-to-heatsink, and heatsink-to-ambient thermal resistances:

$R_{th,jc}$ = .45 (IGBTs) // .92 (diodes) = 0.3022°C/W (Appendix I)

$R_{th,ch}$ = .24°C/W (Appendix I)

$R_{th,ha}$ = 0.24°C/W [19]

Note that $R_{th,jc}$ from the datasheet in Appendix I lists a separate IGBT and diode thermal resistance. According to [20], a parallel connection calculation can be made to determine the overall thermal resistance. This is reflected above. With all the information now gathered, I was able to determine a calculated junction temperature for the IGBT, shown in Fig. 16.

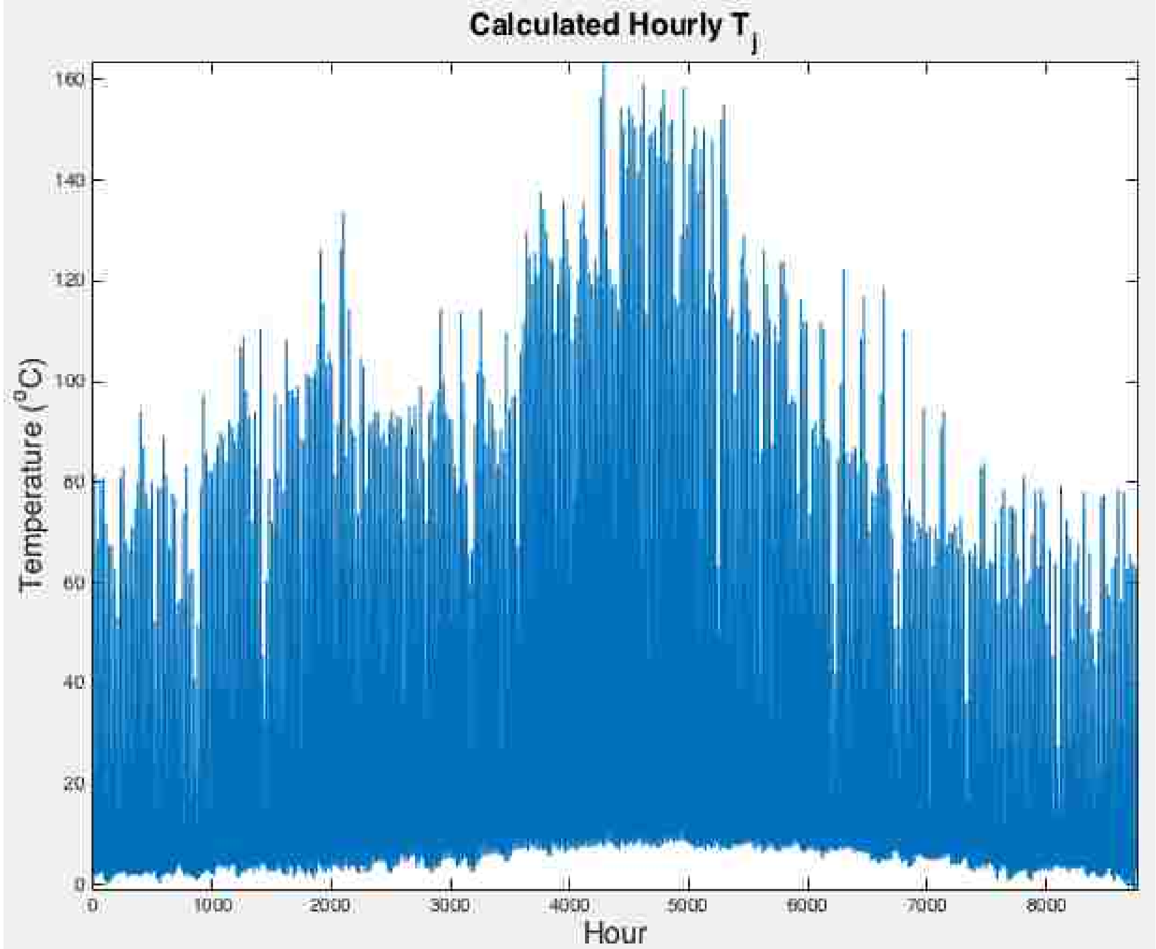


Figure 16. Calculated IGBT Junction Temperature, Case 1.

With the junction temperature calculated, I could now perform the rainflow analysis. As with the earlier calculations, I needed to trim ΔT_j values to no less than 0.001°C and T_{jm} values to no less than 10.9°C. The calculated and trimmed T_{jm} values are shown in Fig. 17.

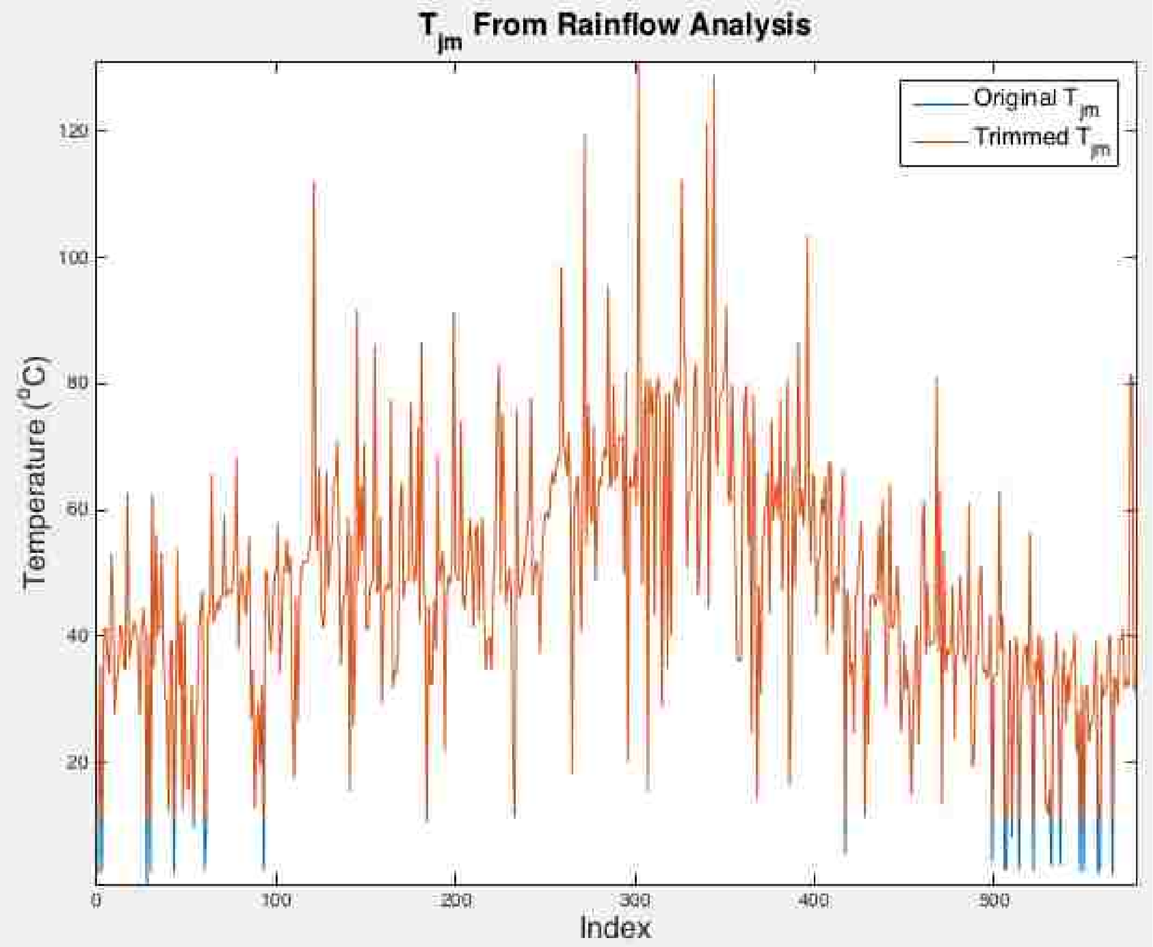


Figure 17. Original and Trimmed T_{jm} for Case 1.

In Fig. 17, you can see just a few data points with values for T_{jm} less than 10.9°C have been trimmed.

With the rainflow analysis complete, I was able to utilize Miner's rule to determine the lifetime of the IGBT: 29.10 years. In contrast to the wildly high values for lifetimes in Chapters 3 and 4, this is a very reasonable lifetime.

5.3 Case 2 Lifetime Estimation.

Next, I performed the lifetime analysis for the second case, where an irradiance greater than 50% allows for max power to be output from the inverter. To accomplish this, I would have to modify some values - I chose to double the irradiance data and clip it to 1. The new equivalent irradiance plot is shown in Fig. 18.

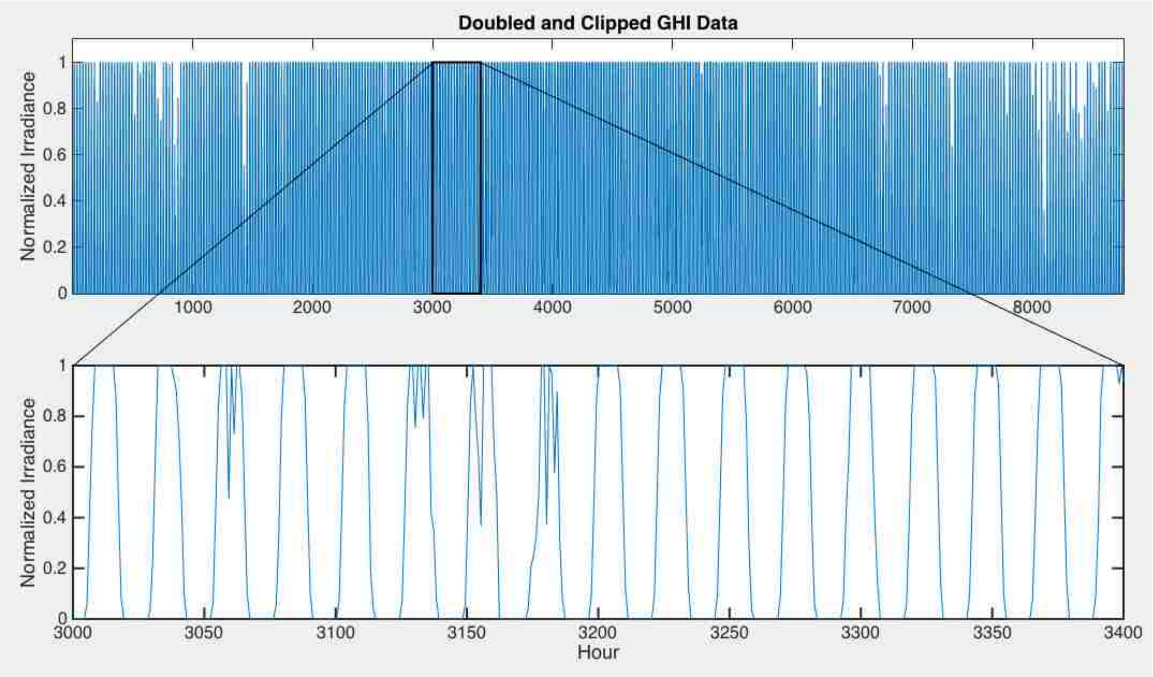


Figure 18. Doubled and Clipped Irradiance Data for Case 2.

Notice above that much of the data is clipped at 1. This is because any value from the original GHI greater than .5 is clipped. Taking the product of the irradiance and load again yields the new normalized power output, shown in Fig. 19.

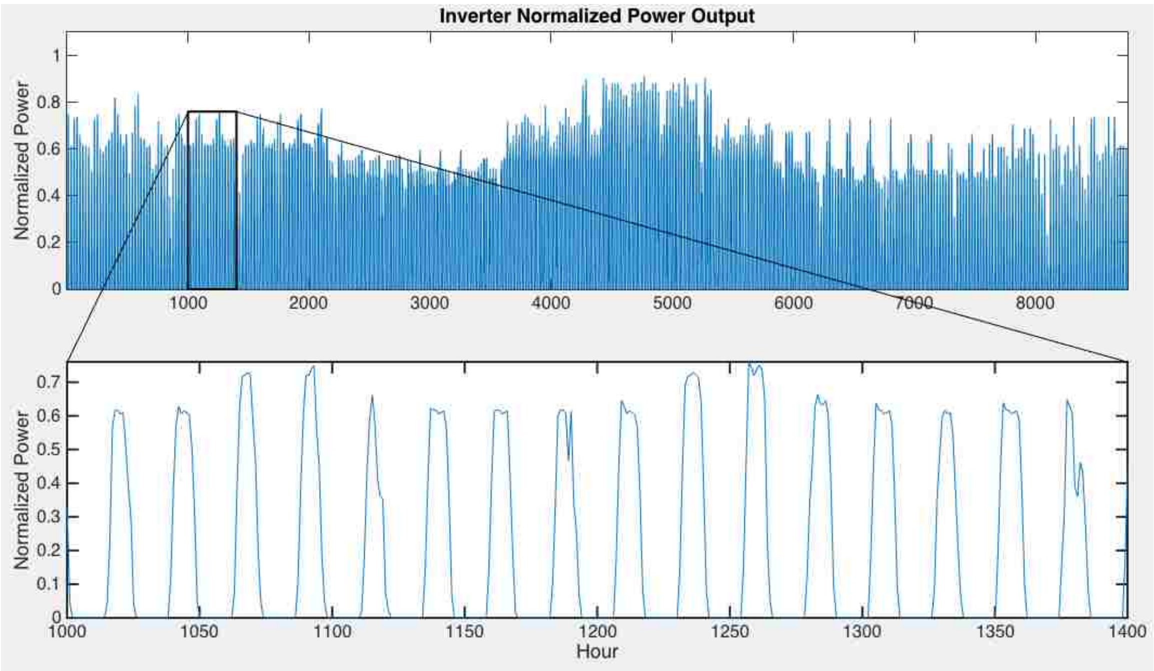


Figure 19. Normalized Hourly Power Output Profile, Case 2.

With the normalized power output calculated, I could now follow the same procedure as in case one to determine the IGBT junction temperature over the course of the year. This is shown in Fig. 20.

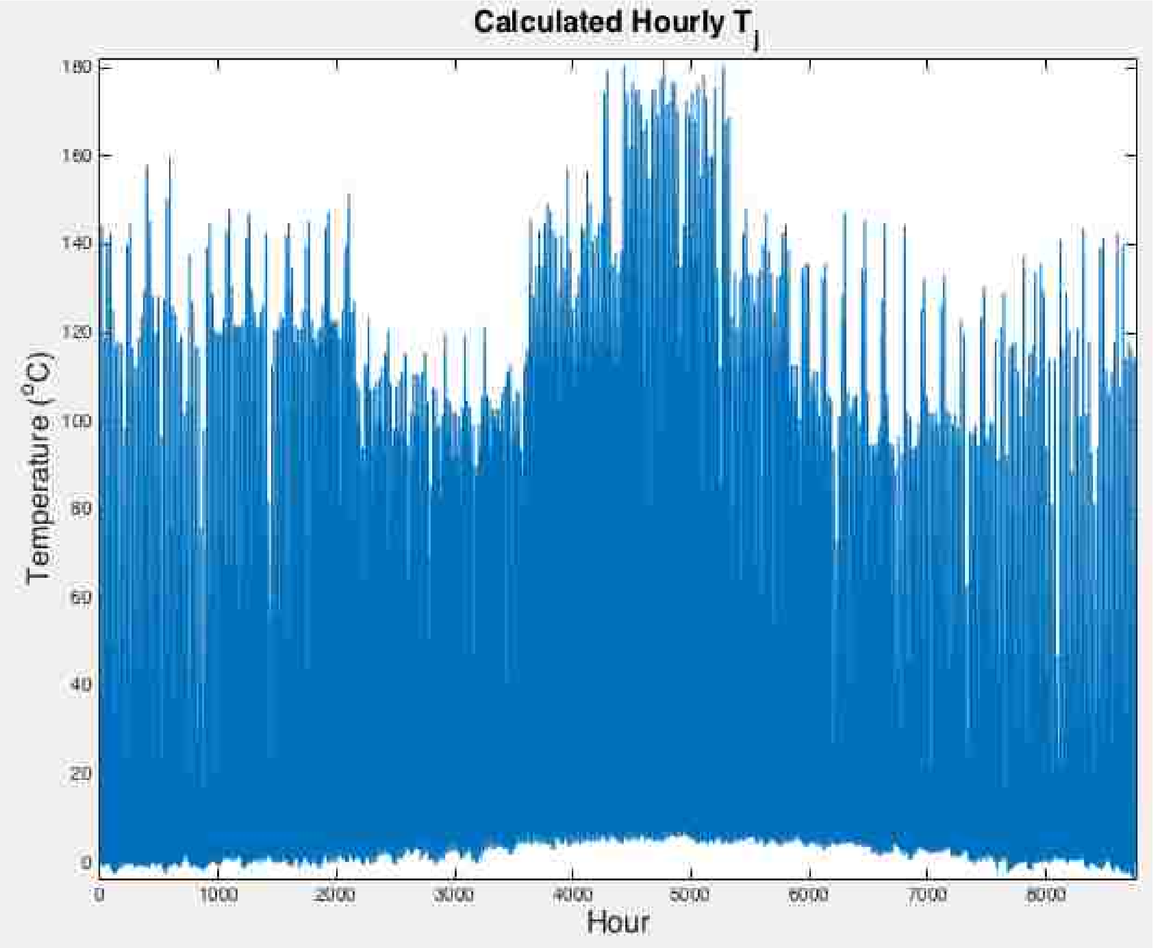


Figure 20. Calculated IGBT Junction Temperature, Case 2.

Note in Fig. 20 that there are temperatures above 180°C at the junction. This is actually problematic, as the maximum junction temperature allowed by the manufacturer is 175°C. Regardless, I proceeded in order to see what kind of lifetime would be calculated for this temperature history.

As before, I trimmed ΔT_j values to no less than 0.001°C and T_{jm} values to no less than 10.9°C. in order prevent overflow errors. The calculated and trimmed T_{jm} values are shown in Fig. 21.

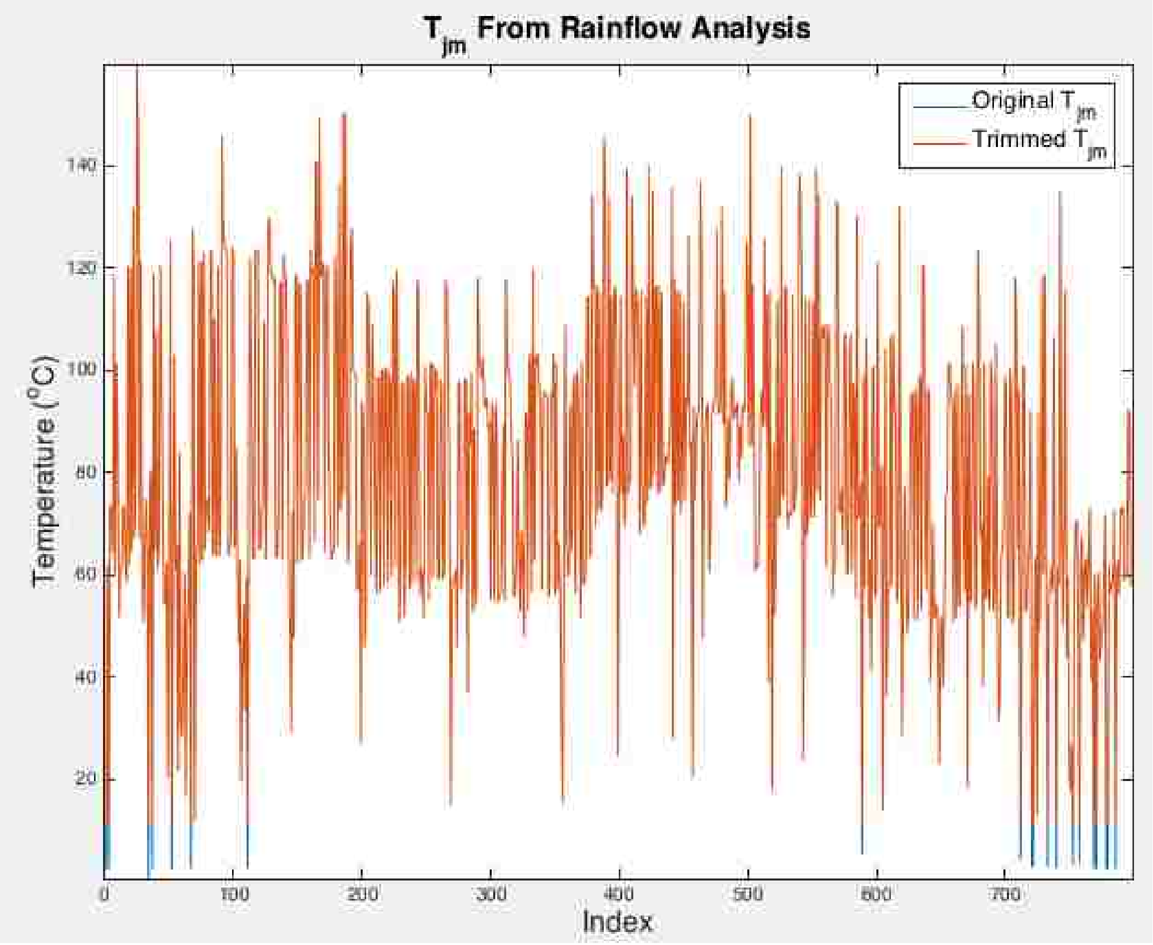


Figure 21. Original and Trimmed T_{jm} for Case 2.

In Fig. 21, you can see just a few data points with values for T_{jm} less than 10.9°C have been trimmed.

With the rainflow analysis complete, I was able to utilize Miner's rule again to determine the lifetime of the IGBT: 4.49 years. Note that this is less than the lifetime that was calculated from case one, which makes sense as the overall output of the inverter, and consequently the IGBT, was increased here in case two.

CHAPTER 6 – SYSTEM RELIABILITY

If I let the failure rates determined in sections 5.2 and 5.3, 29.10 and 4.49 years, respectively, be the mean time to failure (MTTF) for an individual IGBT in a PV inverter, I can calculate reliability numbers for the group of IGBTs in the inverter. Assuming the topology implemented in the inverter is an H-bridge, a very simplified version of which is shown in Fig. 22, there are then a total of four IGBTs.

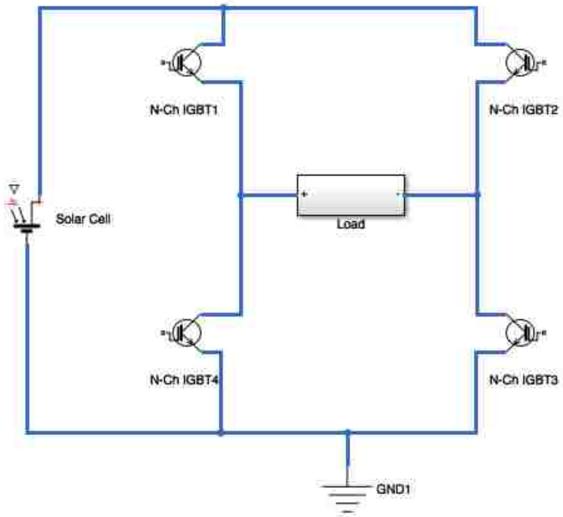


Figure 22. Simplified H-Bridge Topology.

From [21], the inverse of the MTTF (also known as μ) is the failure rate, λ.

$$MTTF = \mu = {}^{1}/_{\lambda} \tag{6}$$

Knowing the failure rate allows for the calculation of the reliability, R:

$$R = e^{-\lambda} \tag{7}$$

Once the reliability of a single IGBT is known, letting the reliability of the other three IGBTs be equivalent allows for the calculation of the reliability of all four IGBTs together. In the case of the H-Bridge, a single IGBT failing would cause the inverter to fail, so the system would be considered to be a series one. The reliability of a system in series is:

$$R_{sys} = \prod_{i=1}^{n} R_i \tag{8}$$

where R_{sys} is the system reliability, and R_i is the reliability of any given component in the system. In the case of a system where the reliabilities are all equal, Eq. 8 reduces to:

$$R_{sys} = R_i^n \tag{9}$$

where R_i is the reliability of each component and n is the number of components in the system. Once the reliability of the four IGBTs is calculated, the failure rate can be calculated by rearranging Eq. 7, and then the MTTF can be calculated for the system by rearranging Eq. 6.

Using Eqs. 6-9 and the calculated lifetimes of the IGBTs from case one and two from section 5.2 and 5.3, the system MTTFs are calculated, with intermediate values shown in Table 4.

Table 4. Intermediate Values Leading to System MTTF for Case 1 and 2.

Parameter	Case 1	Case 2
$\mathrm{MTTF}_{\mathrm{comp}}$ (years)	29.1028	4.4908
λ_{comp}	0.0344	0.2227
R_{comp}	0.9662	0.8004
R_{sys}	0.8716	0.4104
λ_{sys}	0.1374	0.8907
$\mathbf{MTTF}_{\mathbf{sys}}$ (years)	**7.2757**	**1.1227**

Table 4, then, indicates that the MTTF for case one is just 7.28 years, and for case two just 1.12 years.

CHAPTER 7 – CONCLUSIONS

This manuscript details the modeling that I performed towards estimating lifetimes of IGBTs in PV inverters as affected by ambient temperatures utilizing rainflow analysis and Miner's damage rule. Using hourly temperature data for Albuquerque, NM in 2014, I first estimated lifetimes of the IGBTs based on solely the ambient temperature fluctuations, making the assumption that the mean junction temperature was 100^{o}C. I then considered different levels of attenuation on the original temperature profile as a first approximation in accounting for thermal lag. Next, I considered other T_{jm}'s, specifically 60^{o}C and 80^{o}C, in the calculation of the IGBT lifetime. I presented lifetime calculations for the IGBT at three T_{jm} values for 8 different levels of attenuated temperature profiles.

The next stage of modeling that I presented accounted for actual calculated mean junction temperatures in the rainflow analysis. With these results, I presented lifetimes for the IGBT at all 8 different levels of attenuation on the temperature profile.

The last stage of modeling I presented incorporated calculated temperatures for the junction of the IGBT, as opposed to simply using the ambient temperature as previously done, as well as hourly solar irradiance and load demand over the course of a year. Two cases were presented – one where maximum power output from the inverter occurred only when the irradiance was at its peak in conjunction with the load demand at it's peak, and the other where maximum power output from the inverter occurred when irradiance was greater than 50% and load demand was at its peak. Using data from a real

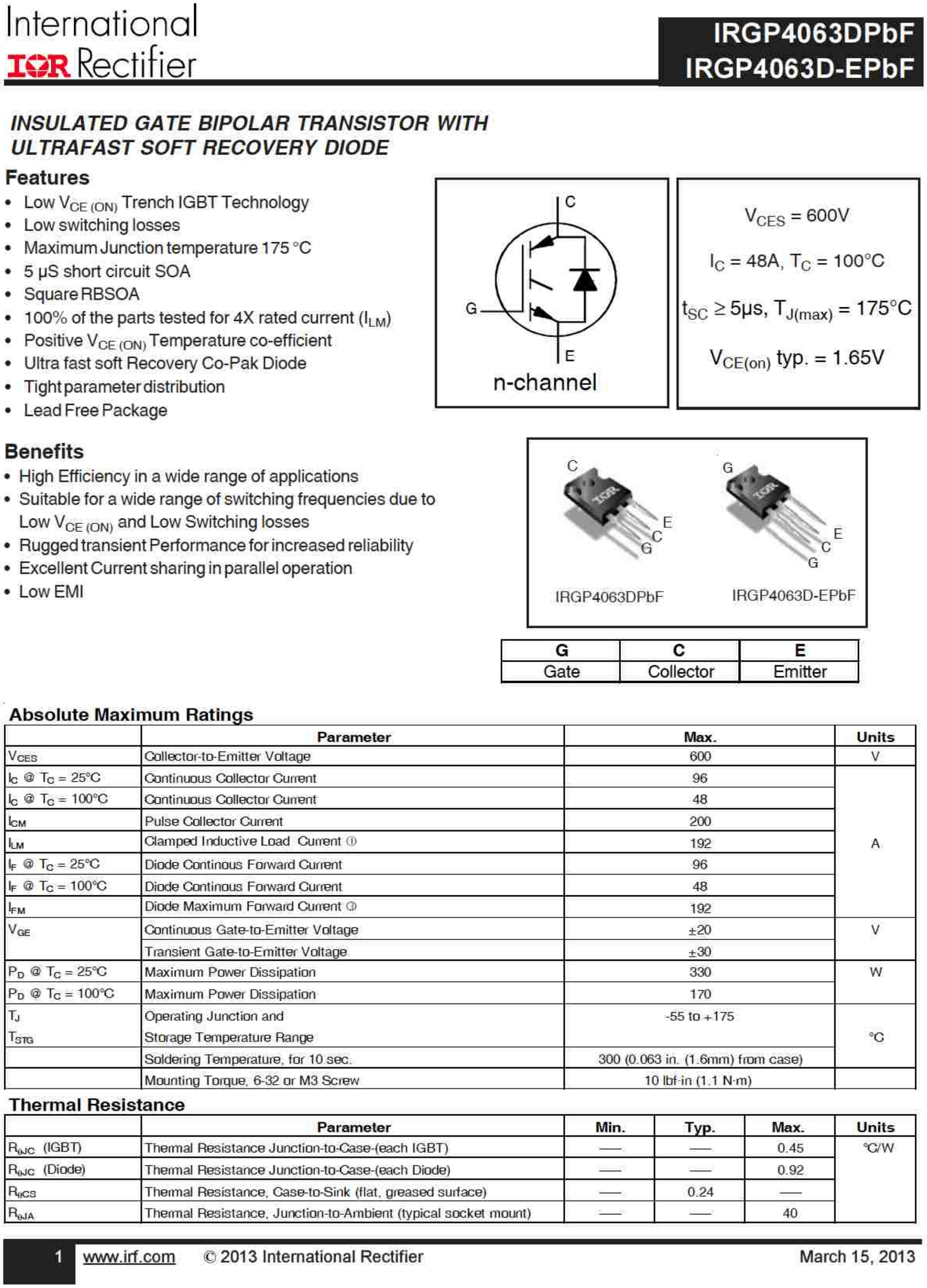
IGBT from International Rectifier, I calculated lifetimes of 29.10 years and 4.49 years for cases 1 and 2, respectively.

Finally, I presented a system reliability analysis of multiple IGBTs in a presumed H-bridge configuration in the PV inverter. Assuming that the mean time to failure for an IGBT in cases 1 and 2 is the lifetime, I calculated the component failure rate and reliability. Knowing that the IGBTs are the same, I calculated a system reliability, which allowed for the backward calculation of a mean time to failure for the whole system. For case 1, I determined a system MTTF of 7.27 years; for case 2, I determined a MTTF of just 1.12 years.

It is important to note that the basic building block of this modeling was the descriptive model, Eq. 1, which was presented in [9]. This model was generated for a specific IGBT at specific conditions. The validity of model was not verified for values of $\Delta T_j < 30K$ nor $\Delta T_j > 80K$, both bounds that I surpassed here. Actual testing beyond these bounds is recommended to verify my modeling.

Future work in this area could also include the utilization of Monte Carlo simulations order to estimate what exactly might occur as well as how likely any situation is to occur. Greater accuracy in lifetime estimates is of the utmost importance when financials are considered. A real-world example of why accurate information is important came out of Spain in 2008, where substantial losses were incurred as a result of overoptimistic calculations for PV LCOE, which include costs such as repairs and replacements [1].

APPENDIX I – IGBT DATASHEET



International IOR Rectifier

IRGP4063DPbF
IRGP4063D-EPbF

INSULATED GATE BIPOLAR TRANSISTOR WITH ULTRAFAST SOFT RECOVERY DIODE

Features

- Low $V_{CE\ (ON)}$ Trench IGBT Technology
- Low switching losses
- Maximum Junction temperature 175 °C
- 5 µS short circuit SOA
- Square RBSOA
- 100% of the parts tested for 4X rated current (I_{LM})
- Positive $V_{CE\ (ON)}$ Temperature co-efficient
- Ultra fast soft Recovery Co-Pak Diode
- Tight parameter distribution
- Lead Free Package

V_{CES} = 600V

I_C = 48A, T_C = 100°C

$t_{SC} \geq 5\mu s$, $T_{J(max)}$ = 175°C

$V_{CE(on)}$ typ. = 1.65V

Benefits

- High Efficiency in a wide range of applications
- Suitable for a wide range of switching frequencies due to Low $V_{CE\ (ON)}$ and Low Switching losses
- Rugged transient Performance for increased reliability
- Excellent Current sharing in parallel operation
- Low EMI

G	C	E
Gate	Collector	Emitter

Absolute Maximum Ratings

	Parameter	Max.	Units
V_{CES}	Collector-to-Emitter Voltage	600	V
I_C @ T_C = 25°C	Continuous Collector Current	96	A
I_C @ T_C = 100°C	Continuous Collector Current	48	
I_{CM}	Pulse Collector Current	200	
I_{LM}	Clamped Inductive Load Current ①	192	
I_F @ T_C = 25°C	Diode Continous Forward Current	96	
I_F @ T_C = 100°C	Diode Continous Forward Current	48	
I_{FM}	Diode Maximum Forward Current ③	192	
V_{GE}	Continuous Gate-to-Emitter Voltage	±20	V
	Transient Gate-to-Emitter Voltage	±30	
P_D @ T_C = 25°C	Maximum Power Dissipation	330	W
P_D @ T_C = 100°C	Maximum Power Dissipation	170	
T_J T_{STG}	Operating Junction and Storage Temperature Range	-55 to +175	°C
	Soldering Temperature, for 10 sec.	300 (0.063 in. (1.6mm) from case)	
	Mounting Torque, 6-32 or M3 Screw	10 lbf·in (1.1 N·m)	

Thermal Resistance

	Parameter	Min.	Typ.	Max.	Units
$R_{\theta JC}$ (IGBT)	Thermal Resistance Junction-to-Case-(each IGBT)	—	—	0.45	°C/W
$R_{\theta JC}$ (Diode)	Thermal Resistance Junction-to-Case-(each Diode)	—	—	0.92	
$R_{\theta CS}$	Thermal Resistance, Case-to-Sink (flat, greased surface)	—	0.24	—	
$R_{\theta JA}$	Thermal Resistance, Junction-to-Ambient (typical socket mount)	—	—	40	

1 www.irf.com © 2013 International Rectifier March 15, 2013

Electrical Characteristics @ T_J = 25°C (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions	Ref.Fig
$V_{(BR)CES}$	Collector-to-Emitter Breakdown Voltage	600	—	—	V	V_{GE} = 0V, I_C = 150μA ④	CT6
$\Delta V_{(BR)CES}/\Delta T_J$	Temperature Coeff. of Breakdown Voltage	—	0.30	—	V/°C	V_{GE} = 0V, I_C = 1mA (25°C-175°C)	CT6
$V_{CE(on)}$	Collector-to-Emitter Saturation Voltage	—	1.65	2.14	V	I_C = 48A, V_{GE} = 15V, T_J = 25°C	5,6,7
		—	2.0	—		I_C = 48A, V_{GE} = 15V, T_J = 150°C	9,10,11
		—	2.05	—		I_C = 48A, V_{GE} = 15V, T_J = 175°C	
$V_{GE(th)}$	Gate Threshold Voltage	4.0	—	6.5	V	V_{CE} = V_{GE}, I_C = 1.4mA	9, 10,
$\Delta V_{GE(th)}/\Delta TJ$	Threshold Voltage temp. coefficient	—	-21	—	mV/°C	V_{CE} = V_{GE}, I_C = 1.0mA (25°C - 175°C)	11, 12
gfe	Forward Transconductance	—	32	—	S	V_{CE} = 50V, I_C = 48A, PW = 80μs	
I_{CES}	Collector-to-Emitter Leakage Current	—	1.0	150	μA	V_{GE} = 0V, V_{CE} = 600V	
		—	450	1000		V_{GE} = 0V, V_{CE} = 600V, T_J = 175°C	
V_{FM}	Diode Forward Voltage Drop	—	1.95	2.91	V	I_F = 48A	8
		—	1.45	—		I_F = 48A, T_J = 175°C	
I_{GES}	Gate-to-Emitter Leakage Current	—	—	±100	nA	V_{GE} = ±20V	

Switching Characteristics @ T_J = 25°C (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions	Ref.Fig
Q_g	Total Gate Charge (turn-on)	—	95	140	nC	I_C = 48A	24
Q_{ge}	Gate-to-Emitter Charge (turn-on)	—	28	42		V_{GE} = 15V	CT1
Q_{gc}	Gate-to-Collector Charge (turn-on)	—	35	53		V_{CC} = 400V	
E_{on}	Turn-On Switching Loss	—	625	1141	μJ	I_C = 48A, V_{CC} = 400V, V_{GE} = 15V	CT4
E_{off}	Turn-Off Switching Loss	—	1275	1481		R_G = 10Ω, L = 200μH, L_S = 150nH, T_J = 25°C	
E_{total}	Total Switching Loss	—	1900	2622		Energy losses include tail & diode reverse recovery	
$t_{d(on)}$	Turn-On delay time	—	60	78	ns	I_C = 48A, V_{CC} = 400V, V_{GE} = 15V	CT4
t_r	Rise time	—	40	56		R_G = 10Ω, L = 200μH, L_S = 150nH, T_J = 25°C	
$t_{d(off)}$	Turn-Off delay time	—	145	176			
t_f	Fall time	—	35	46			
E_{on}	Turn-On Switching Loss	—	1625	—	μJ	I_C = 48A, V_{CC} = 400V, V_{GE}=15V	13, 15
E_{off}	Turn-Off Switching Loss	—	1585	—		R_G=10Ω, L=200μH, L_S=150nH, T_J = 175°C ④	CT4
E_{total}	Total Switching Loss	—	3210	—		Energy losses include tail & diode reverse recovery	WF1, WF2
$t_{d(on)}$	Turn-On delay time	—	55	—	ns	I_C = 48A, V_{CC} = 400V, V_{GE} = 15V	14, 16
t_r	Rise time	—	45	—		R_G = 10Ω, L = 200μH, L_S = 150nH	CT4
$t_{d(off)}$	Turn-Off delay time	—	165	—		T_J = 175°C	WF1
t_f	Fall time	—	45	—			WF2
C_{ies}	Input Capacitance	—	3025	—	pF	V_{GE} = 0V	23
C_{oes}	Output Capacitance	—	245	—		V_{CC} = 30V	
C_{res}	Reverse Transfer Capacitance	—	90	—		f = 1.0Mhz	
RBSOA	Reverse Bias Safe Operating Area	FULL SQUARE				T_J = 175°C, I_C = 192A V_{CC} = 480V, Vp =600V Rg = 10Ω, V_{GE} = +15V to 0V	4 CT2
SCSOA	Short Circuit Safe Operating Area	5	—	—	μs	V_{CC} = 400V, Vp =600V Rg = 10Ω, V_{GE} = +15V to 0V	22, CT3 WF4
Erec	Reverse Recovery Energy of the Diode	—	845	—	μJ	T_J = 175°C	17, 18, 19
t_{rr}	Diode Reverse Recovery Time	—	115	—	ns	V_{CC} = 400V, I_F = 48A	20, 21
I_{rr}	Peak Reverse Recovery Current	—	40	—	A	V_{GE} = 15V, Rg = 10Ω, L =200μH, L_s = 150nH	WF3

Notes:

① V_{CC} = 80% (V_{CES}), V_{GE} = 20V, L = 200μH, R_G = 10Ω.

② This is only applied to TO-247AC package.

③ Pulse width limited by max. junction temperature.

④ Refer to AN-1086 for guidelines for measuring $V_{(BR)CES}$ safely.

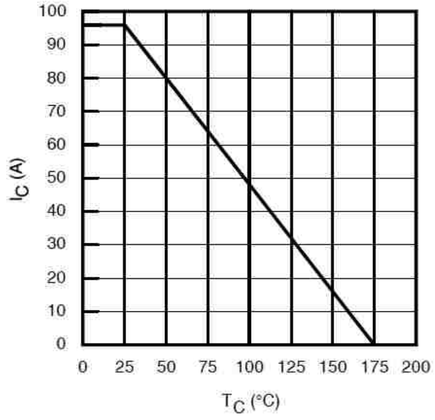


Fig. 1 - Maximum DC Collector Current vs. Case Temperature

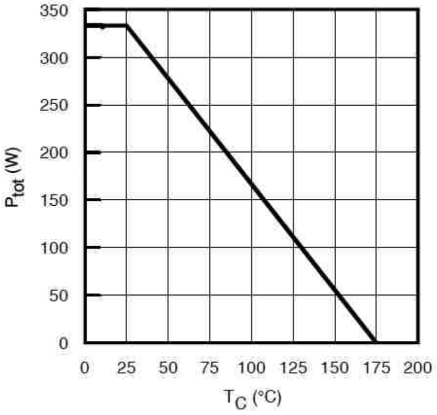


Fig. 2 - Power Dissipation vs. Case Temperature

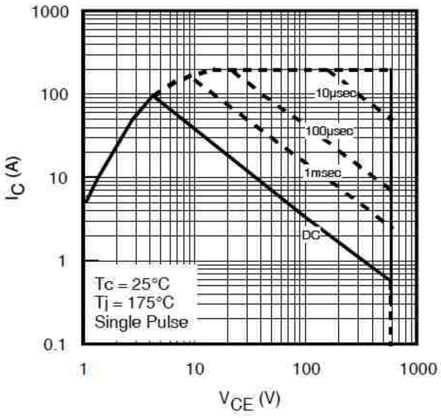


Fig. 3 - Forward SOA
$T_C = 25°C$, $T_J \leq 175°C$; $V_{GE} = 15V$

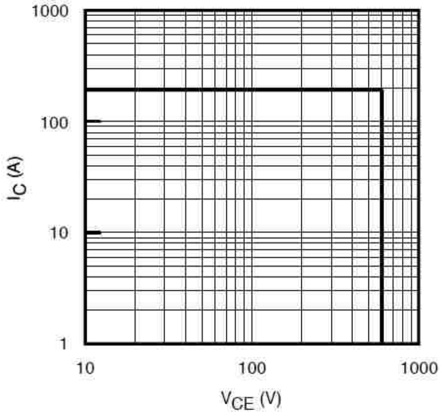


Fig. 4 - Reverse Bias SOA
$T_J = 175°C$; $V_{GE} = 15V$

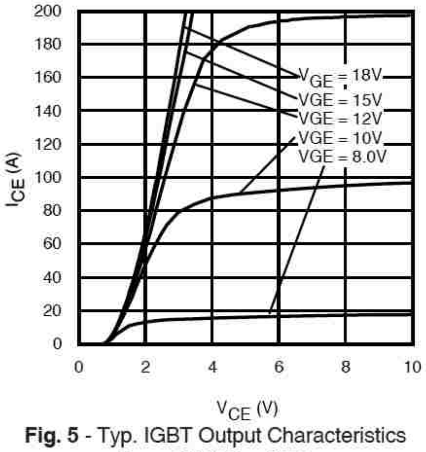


Fig. 5 - Typ. IGBT Output Characteristics
$T_J = -40°C$; tp = 80μs

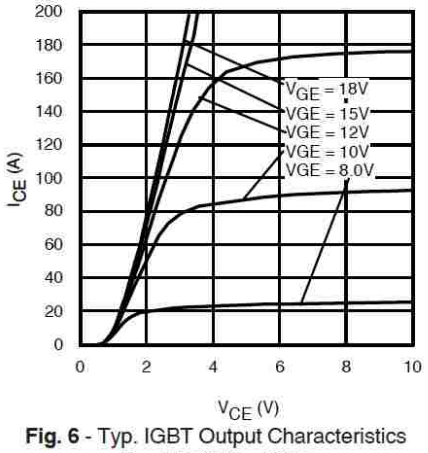


Fig. 6 - Typ. IGBT Output Characteristics
$T_J = 25°C$; tp = 80μs

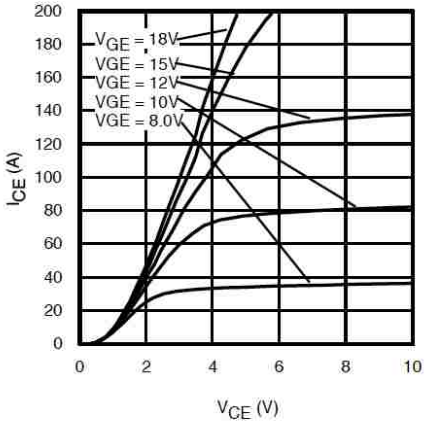


Fig. 7 - Typ. IGBT Output Characteristics
T_J = 175°C; tp = 80µs

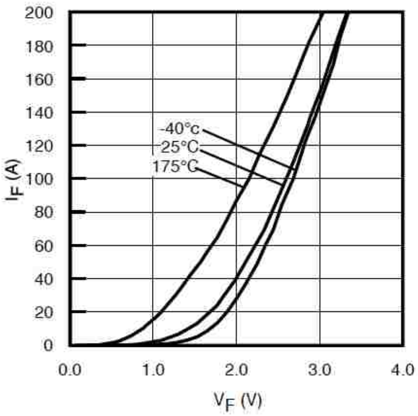


Fig. 8 - Typ. Diode Forward Characteristics
tp = 80µs

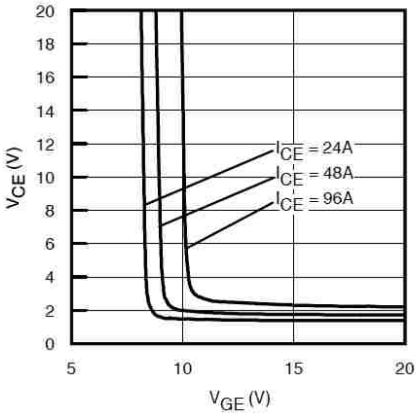


Fig. 9 - Typical V_{CE} vs. V_{GE}
T_J = -40°C

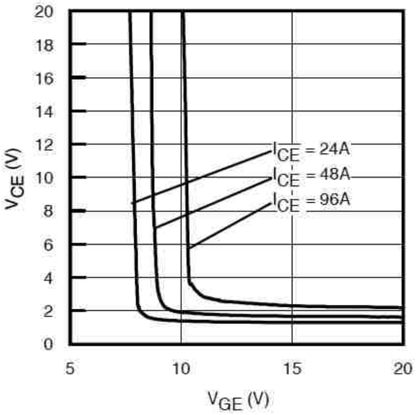


Fig. 10 - Typical V_{CE} vs. V_{GE}
T_J = 25°C

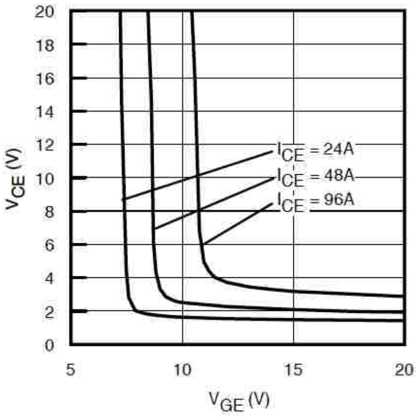


Fig. 11 - Typical V_{CE} vs. V_{GE}
T_J = 175°C

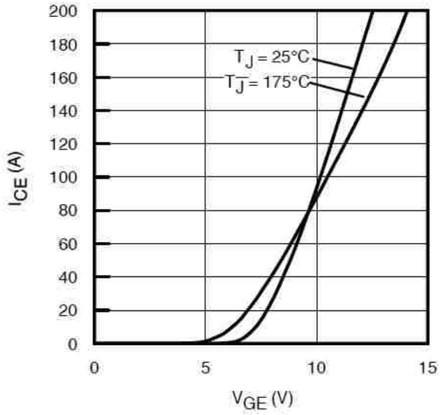


Fig. 12 - Typ. Transfer Characteristics
V_{CE} = 50V; tp = 10µs

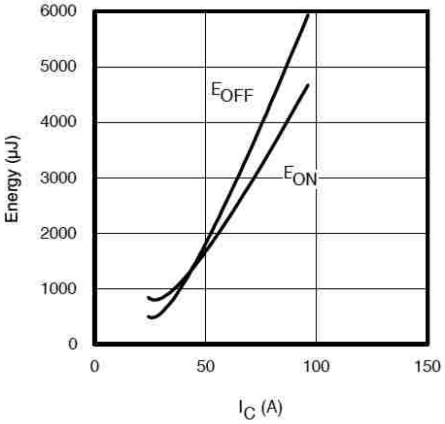


Fig. 13 - Typ. Energy Loss vs. I_C
T_J = 175°C; L = 200µH; V_{CE} = 400V, R_G = 10Ω; V_{GE} = 15V

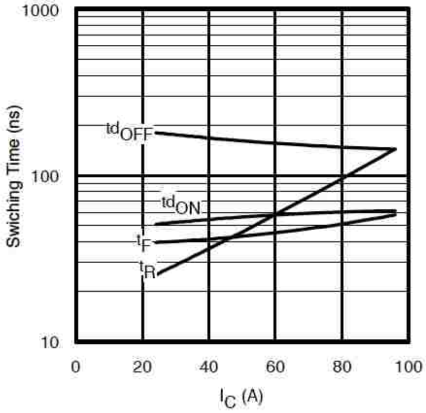


Fig. 14 - Typ. Switching Time vs. I_C
T_J = 175°C; L = 200µH; V_{CE} = 400V, R_G = 10Ω; V_{GE} = 15V

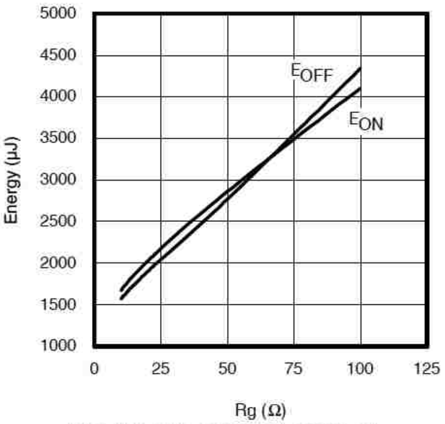


Fig. 15 - Typ. Energy Loss vs. R_G
T_J = 175°C; L = 200µH; V_{CE} = 400V, I_{CE} = 48A; V_{GE} = 15V

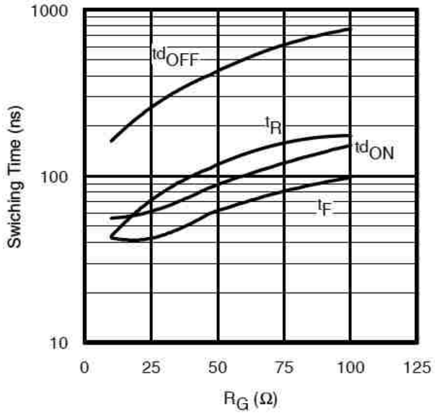


Fig. 16 - Typ. Switching Time vs. R_G
T_J = 175°C; L = 200µH; V_{CE} = 400V, I_{CE} = 48A; V_{GE} = 15V

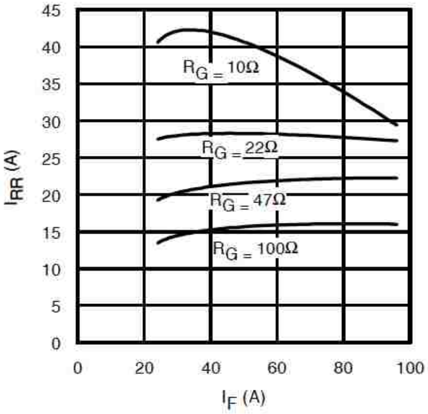


Fig. 17 - Typ. Diode I_{RR} vs. I_F
T_J = 175°C

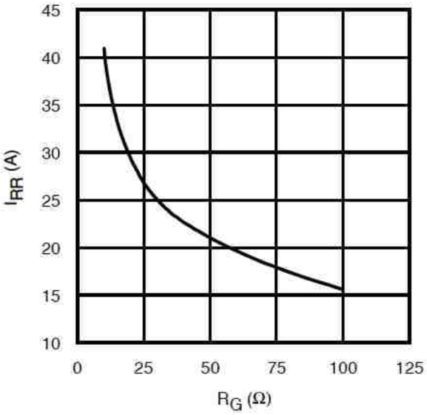


Fig. 18 - Typ. Diode I_{RR} vs. R_G
T_J = 175°C

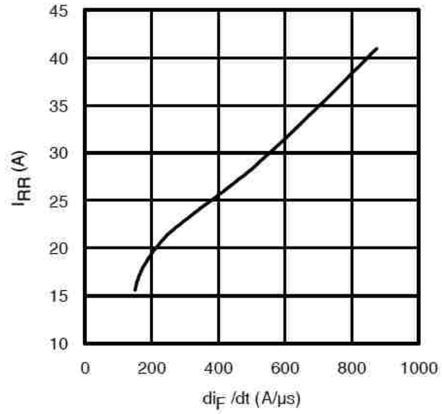


Fig. 19 - Typ. Diode I_{RR} vs. di_F/dt
V_{CC} = 400V; V_{GE} = 15V; I_F = 48A; T_J = 175°C

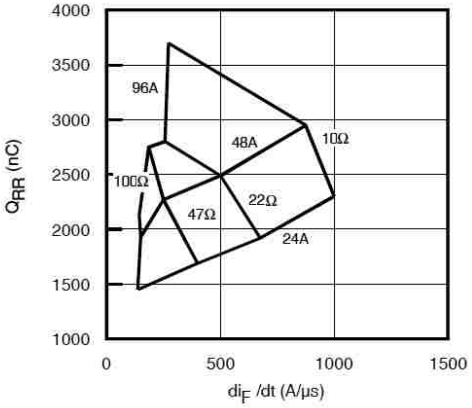


Fig. 20 - Typ. Diode Q_{RR} vs. di_F/dt
V_{CC} = 400V; V_{GE} = 15V; T_J = 175°C

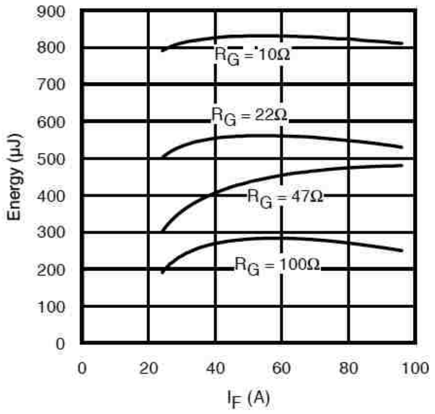


Fig. 21 - Typ. Diode E_{RR} vs. I_F
T_J = 175°C

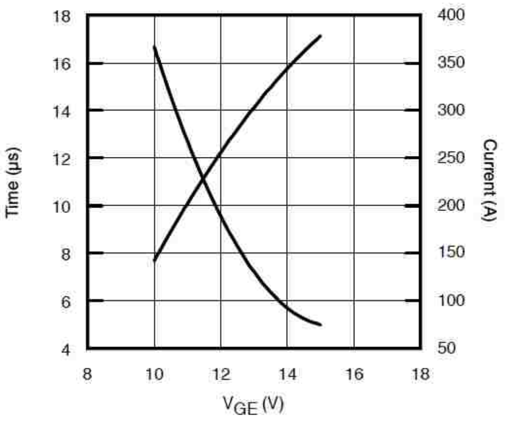


Fig. 22 - V_{GE} vs. Short Circuit Time
V_{CC} = 400V; T_C = 25°C

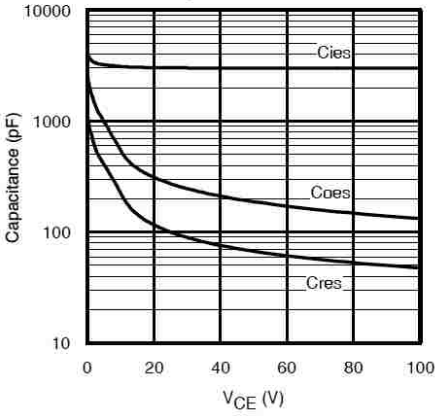


Fig. 23 - Typ. Capacitance vs. V_{CE}
V_{GE}= 0V; f = 1MHz

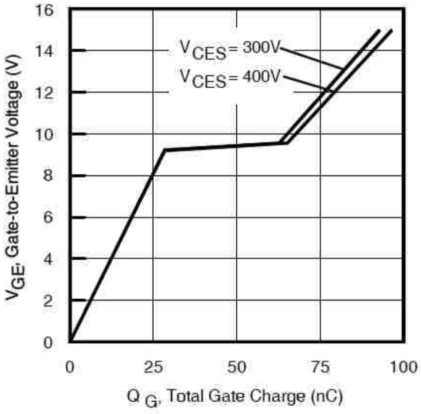
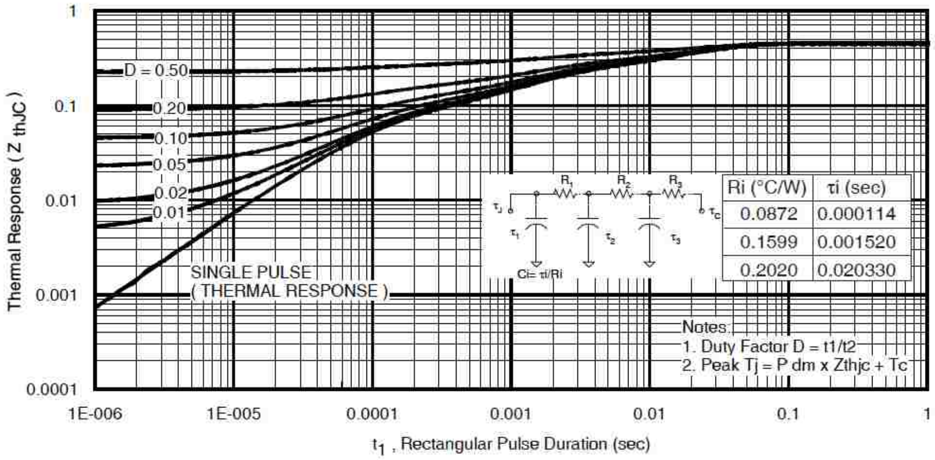
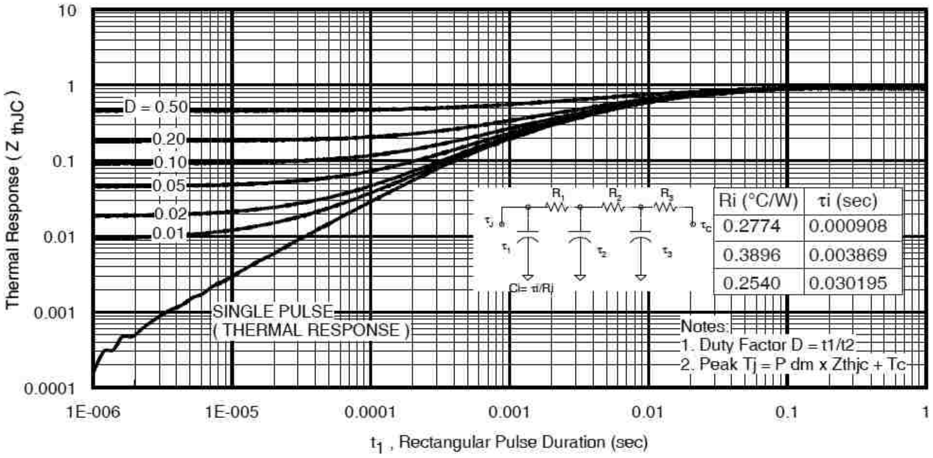


Fig. 24 - Typical Gate Charge vs. V_{GE}
I_{CE} = 48A; L = 600μH



Ri (°C/W)	τi (sec)
0.0872	0.000114
0.1599	0.001520
0.2020	0.020330

Fig 25. Maximum Transient Thermal Impedance, Junction-to-Case (IGBT)



Ri (°C/W)	τi (sec)
0.2774	0.000908
0.3896	0.003869
0.2540	0.030195

Fig. 26. Maximum Transient Thermal Impedance, Junction-to-Case (DIODE)

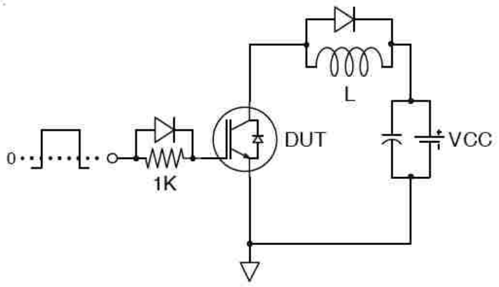


Fig.C.T.1 - Gate Charge Circuit (turn-off)

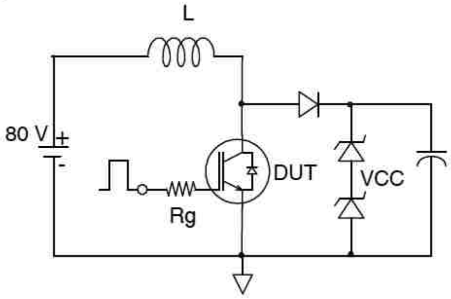


Fig.C.T.2 - RBSOA Circuit

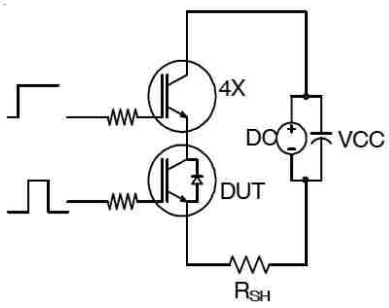


Fig.C.T.3 - S.C. SOA Circuit

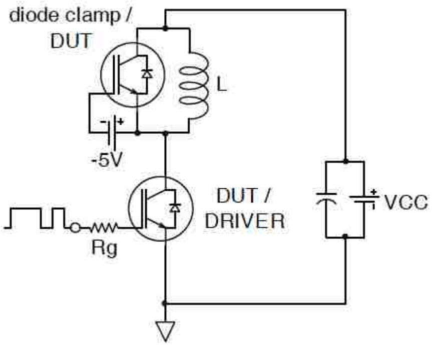


Fig.C.T.4 - Switching Loss Circuit

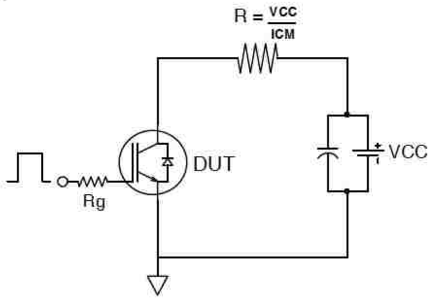


Fig.C.T.5 - Resistive Load Circuit

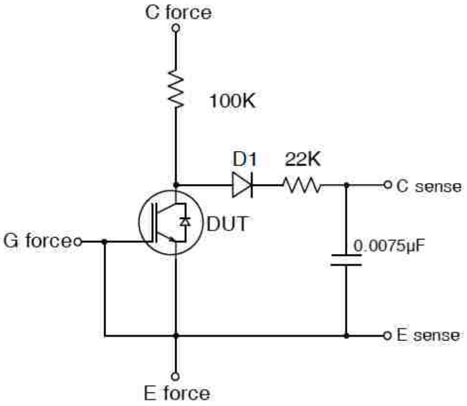


Fig.C.T.6 - BVCES Filter Circuit

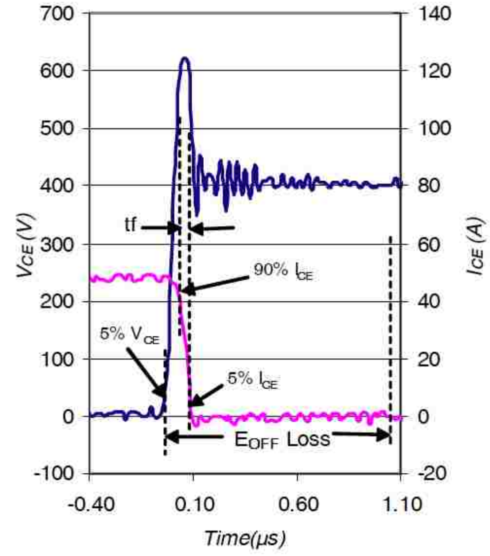


Fig. WF1 - Typ. Turn-off Loss Waveform
@ T_J = 175°C using Fig. CT.4

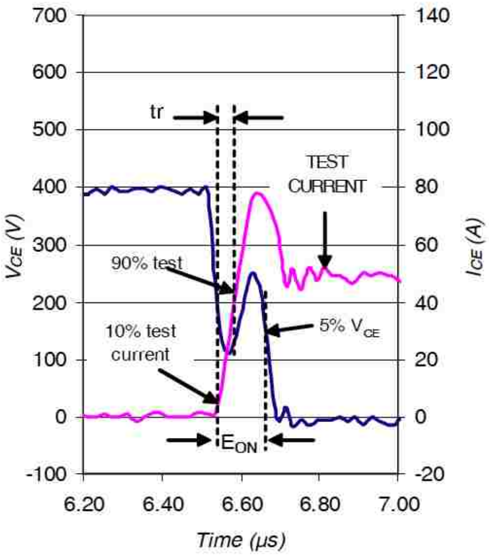


Fig. WF2 - Typ. Turn-on Loss Waveform
@ T_J = 175°C using Fig. CT.4

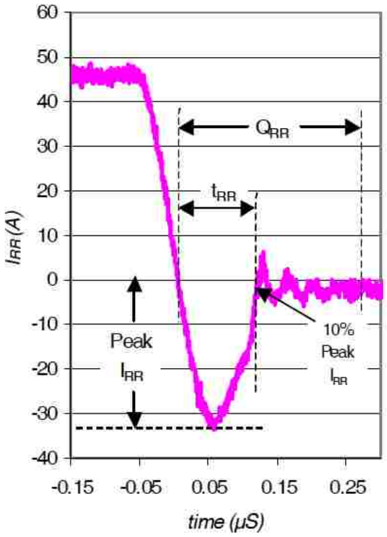


Fig. WF3 - Typ. Diode Recovery Waveform
@ T_J = 175°C using Fig. CT.4

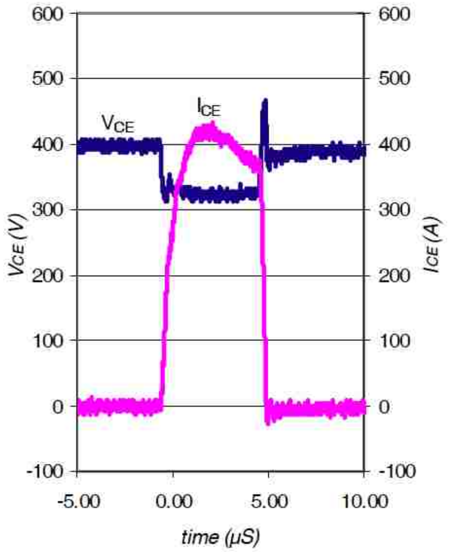
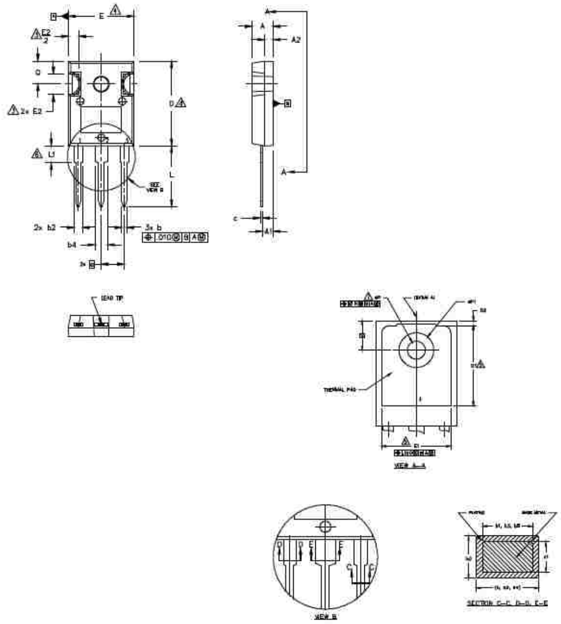


Fig. WF4 - Typ. S.C. Waveform
@ T_J = 25°C using Fig. CT.3

TO-247AC Package Outline

Dimensions are shown in millimeters (inches)



NOTES:

1. DIMENSIONING AND TOLERANCING AS PER ASME Y14.5M 1994.
2. DIMENSIONS ARE SHOWN IN INCHES.
3. CONTOUR OF SLOT OPTIONAL.
4. DIMENSION D & E DO NOT INCLUDE MOLD FLASH. MOLD FLASH SHALL NOT EXCEED .005" (0.127) PER SIDE. THESE DIMENSIONS ARE MEASURED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY.
5. THERMAL PAD CONTOUR OPTIONAL WITHIN DIMENSIONS D1 & E1.
6. LEAD FINISH UNCONTROLLED IN L1.
7. øP TO HAVE A MAXIMUM DRAFT ANGLE OF 1.5 ° TO THE TOP OF THE PART WITH A MAXIMUM HOLE DIAMETER OF .154 INCH.
8. OUTLINE CONFORMS TO JEDEC OUTLINE TO-247AC .

SYMBOL	DIMENSIONS				NOTES
	INCHES		MILLIMETERS		
	MIN.	MAX.	MIN.	MAX.	
A	.183	.209	4.65	5.31	
A1	.087	.102	2.21	2.59	
A2	.059	.098	1.50	2.49	
b	.039	.055	0.99	1.40	
b1	.039	.053	0.99	1.35	
b2	.065	.094	1.65	2.39	
b3	.065	.092	1.65	2.34	
b4	.102	.135	2.59	3.43	
b5	.102	.133	2.59	3.38	
c	.015	.035	0.38	0.89	
c1	.015	.033	0.38	0.84	
D	.776	.815	19.71	20.70	4
D1	.515	–	13.08	–	5
D2	.020	.053	0.51	1.35	
E	.602	.625	15.29	15.87	4
E1	.530	–	13.46	–	
E2	.178	.216	4.52	5.49	
e	.215 BSC		5.46 BSC		
øk	.010		0.25		
L	.559	.634	14.20	16.10	
L1	.146	.169	3.71	4.29	
øP	.140	.144	3.56	3.66	
øP1	–	.291	–	7.39	
Q	.209	.224	5.31	5.69	
S	.217 BSC		5.51 BSC		

LEAD ASSIGNMENTS

HEXFET

1.– GATE
2.– DRAIN
3.– SOURCE
4.– DRAIN

IGBTs, CoPACK

1.– GATE
2.– COLLECTOR
3.– EMITTER
4.– COLLECTOR

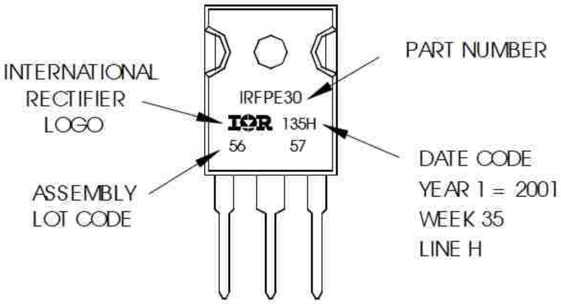
DIODES

1.– ANODE/OPEN
2.– CATHODE
3.– ANODE

TO-247AC Part Marking Information

EXAMPLE: THIS IS AN IRFPE30
WITH ASSEMBLY
LOT CODE 5657
ASSEMBLED ON WW 35, 2001
IN THE ASSEMBLY LINE "H"

Note: "P" in assembly line position
indicates "Lead-Free"

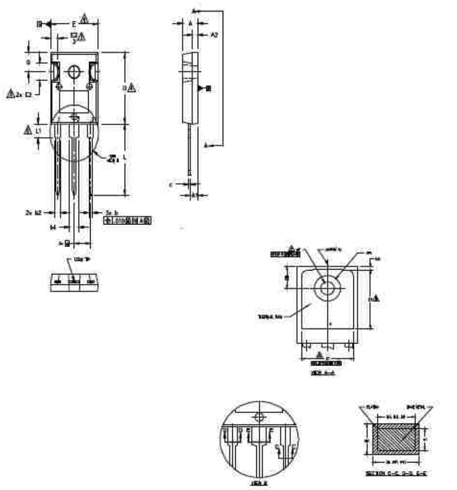


TO-247AC package is not recommended for Surface Mount Application.

Note: For the most current drawing please refer to IR website at http://www.irf.com/package/

TO-247AD Package Outline

Dimensions are shown in millimeters (inches)



NOTES:

1. DIMENSIONING AND TOLERANCING AS PER ASME Y14.5M 1994.
2. DIMENSIONS ARE SHOWN IN INCHES.
3. CONTOUR OF SLOT OPTIONAL.
4. DIMENSION D & E DO NOT INCLUDE MOLD FLASH. MOLD FLASH SHALL NOT EXCEED .005" (0.127) PER SIDE. THESE DIMENSIONS ARE MEASURED AT THE OUTERMOST EXTREMES OF THE PLASTIC BODY.
5. THERMAL PAD CONTOUR OPTIONAL WITHIN DIMENSIONS D1 & E1.
6. LEAD FINISH UNCONTROLLED IN L1.
7. ØP TO HAVE A MAXIMUM DRAFT ANGLE OF 1.5 ° TO THE TOP OF THE PART WITH A MAXIMUM HOLE DIAMETER OF .154 INCH.
8. OUTLINE CONFORMS TO JEDEC OUTLINE TO-247AD.

SYMBOL	DIMENSIONS				NOTES
	INCHES		MILLIMETERS		
	MIN.	MAX.	MIN.	MAX.	
A	.185	.209	4.65	5.31	
A1	.087	.102	2.21	2.59	
A2	.059	.098	1.50	2.49	
b	.039	.056	0.99	1.40	
b1	.039	.053	0.99	1.35	
b2	.065	.094	1.65	2.39	
b3	.065	.092	1.65	2.34	
b4	.102	.135	2.59	3.43	
b5	.102	.133	2.59	3.38	
c	.015	.035	0.38	0.89	
c1	.015	.033	0.38	0.84	
D	.776	.815	19.71	20.70	4
D1	.515	–	13.08	–	5
D2	.020	.053	0.51	1.35	
E	.602	.625	15.29	15.87	4
E1	.530	–	13.46	–	
E2	.178	.216	4.52	5.49	
e	.215 BSC		5.46 BSC		
Øk	.010		0.25		
L	.780	.827	19.81	21.00	
L1	.146	.169	3.71	4.29	
ØP	.140	.144	3.56	3.66	
ØP1	–	.291	–	7.39	
Q	.209	.224	5.31	5.69	
S	.217 BSC		5.51 BSC		

LEAD ASSIGNMENTS

HEXFET

1.- GATE
2.- DRAIN
3.- SOURCE
4.- DRAIN

IGBTs, CoPACK

1.- GATE
2.- COLLECTOR
3.- EMITTER
4.- COLLECTOR

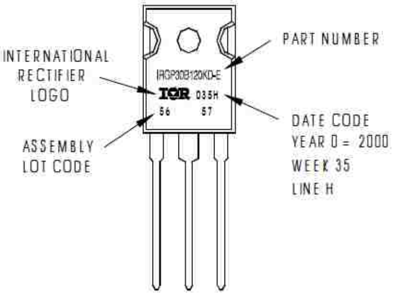
DIODES

1.- ANODE/OPEN
2.- CATHODE
3.- ANODE

TO-247AD Part Marking Information

EXAMPLE: THIS IS AN IRGP30B120KD-E
WITH ASSEMBLY
LOT CODE 5657
ASSEMBLED ON WW 35, 2000
IN THE ASSEMBLY LINE "H"

Note: "P" in assembly line position
indicates "Lead-Free"



TO-247AD package is not recommended for Surface Mount Application.

Note: For the most current drawing please refer to IR website at http://www.irf.com/package/

Data and specifications subject to change without notice.
This product has been designed and qualified for Industrial market.
Qualification Standards can be found on IR's Web site.

International
IOR Rectifier

IR WORLD HEADQUARTERS: 101 N. Sepulveda Blvd.., El Segundo, California 90245, USA Tel: (310) 252-7105
TAC Fax: (310) 252-7903
Visit us at www.irf.com for sales contact information.

APPENDIX II – MATLAB CODE

```
clear all

%import hourly temp and irradiance
excelFile=xlsread('/Users/Richard/Documents/AAAA Grad
School/Thesis/ABQ_hourly_data.xlsx');
hourlyTempData=excelFile(3:8762,11); %ABQ temp 2014

hourlyIrradiance=excelFile(3:8762,9); %global horizontal irradiance...
    %in ABQ in 2014
hourlyIrradiance=hourlyIrradiance/max(hourlyIrradiance); %scales...
    %data to max value of 1


%import hourly load data for ABQ
excelFile=xlsread('/Users/Richard/Documents/AAAA Grad
School/Thesis/hourly-annual_load_profile.xlsx');
hourlyLoad=excelFile(1:365, 2:25); %pulls data and eliminates headers
hourlyLoad=transpose(hourlyLoad);
hourlyLoad=reshape(hourlyLoad, [1, 8760]); %flattens matrix
hourlyLoad=hourlyLoad/max(hourlyLoad); %scales data to max value of 1
hourlyLoad=transpose(hourlyLoad);

%hourly POut
hourlyPOut = hourlyIrradiance.*hourlyLoad; % irradiance-load product

% Get peaks and troughs
sequence=sig2ext(hourlyTempData);

% shift temp data to coincide with max IGBT temp of 150 deg C
shift=150-max(sequence);
sequence=sequence+shift;

%RF
rf=rainflow(sequence);
CycleRate=rf(3,:);   % number of cycles
siga=rf(1,:);   %cycle amplitudes

% calculation of the damage
sigaf=50;   % temp endurance limit
% m=8;         % slope of the curve2
Nk=6e5;      % number of cycles at knee point
damage=sum((CycleRate/Nk).*(siga/sigaf));



%% RF cycle extraction
figure('units','normalized','position',[.1 .15 .8 .75])
rfdemo1(sequence(1:20))
set(gca, 'fontsize', 16)
```

```
%% low pass filter the signal
filterOrder=8; % even integers only, please
cutoff=.08; % Max cutoff=1, which is Nyquist freq -> pi rad/sample
outhi=attenuateTemp(hourlyTempData, filterOrder,cutoff);
z=linspace(0,length(hourlyTempData),length(hourlyTempData));
minX=100;
maxX=400;
figure('units','normalized','position',[.25 .25 .5 .5])
plot(z(minX:maxX), hourlyTempData(minX:maxX), z(minX:maxX),...
    outhi(minX+filterOrder/2:maxX+filterOrder/2)) % note that the...
        % second signal is shifted to overlay
    h=title('Hourly Temperature Data, Albuquerque, NM, 2014');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    h=legend('Original', 'Attenuated');
        h.FontSize=13;
    axis([minX,maxX,-inf,inf]);



%% plot temps full range
% x=linspace(0,length(outhi),length(outhi));
padLead=filterOrder/2;
padTail=length(outhi)-filterOrder/2-1;
% plot(x(padLead:padTail), hourlyTempData, x, outhi)
figure('units','normalized','position',[.1 .25 .8 .6])
plot(z, hourlyTempData, z, outhi(padLead:padTail))
    h=title('Hourly Temperature Data, Albuquerque, NM, 2014');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    h=legend('Original', 'Attenuated');
        h.FontSize=13;

%% plot temps together
figure('units','normalized','position',[.1 .15 .8 .75])
plot(z, hourlyTempData, z, outhi(padLead:padTail));
grid off;
axis([1,length(hourlyTempData),-12,40]);
ylabel('Temperature (^{o}C)');
title('Hourly Temperature Data, Albuquerque, NM, 2014');
set(gca, 'fontsize', 16, 'linewidth', 1);
% ylim([-10 2]);

props.axes1.position = [0.1 0.55 0.85 0.35];
props.axes2.position = [0.1 0.1 0.85 0.35];
props.axes1.fontsize = 16;
props.axes2.fontsize = 16;
props.axes2.linewidth = 2;
props.xlabel = 'Hour';
props.ylabel = 'Temperature (^{o}C)';
```

```
[ax1, ax2] = popout(gcf, 1000, 1400, props);
h=legend('Original', 'Attenuated'); % move legend as appropriate
    h.FontSize=13;
    h.LineWidth=.5;



%% low pass filter the signal
cutoff=.08; % Max cutoff=1, which is Nyquist freq -> pi rad/sample...
    % 0.08 is a good value
fOMin=8;
fOMax=512;
n=log2(fOMax)-log2(fOMin)+1; % # of lines, add 1 because of extra...
    % element inclusive of fOMin
outhi=zeros(n,length(hourlyTempData)+fOMax); % placeholder for plot...
    % lines
filterOrder=zeros(n,1);
i=fOMin;
while i <= fOMax %otherwise will run to fOMax*2
    index=log2(i)-2;
    temp=attenuateTemp(hourlyTempData, i, cutoff);
    if length(temp)<length(outhi)
        diff=length(outhi)-length(temp);
        temp=padarray(temp,[diff/2,0],0);
        temp=transpose(temp);
        outhi(index,:)=temp;
    else outhi(index,:)=transpose(temp);
    end
    filterOrder(index)=i;
    i=i*2;
end
clear temp diff i;

z=linspace(0,length(hourlyTempData),length(hourlyTempData));
minX=1250; %1250-1400 is nice
maxX=1400;
figure('units','normalized','position',[.25 .25 .5 .6])
plot(z(minX:maxX), hourlyTempData(minX:maxX),'Color',[0 0.4470 0.7410])
hold on
for i=1:n
    plot(z(minX:maxX), outhi(i,minX+filterOrder(i)/2+(2^(n+1)-
2^(1+i)):...
        maxX+filterOrder(i)/2+(2^(n+1)-2^(1+i)))) % note that...
        % the second signal is shifted to overlay
end
    h=title('Hourly Attenuated Temperatures');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    h=legend('Original', 'Filter order: 8', 'Filter order: 16',...
        'Filter order: 32','Filter order: 64', 'Filter order: 128',...
        'Filter order: 256', 'Filter order: 512');
    h.FontSize=13;
    axis([minX,maxX,-2,23]);
    hold off
```

```
%% Set up rainflow amplitudes
% original temp profile
rf=rainflow(sequence);
siga=rf(1,:);
sigaT=transpose(siga); % for easy viewing/debugging

% attenuated temp profile
sequence1=sig2ext(outhi(padLead:padTail));
rf1=rainflow(sequence1);
siga1=rf1(1,:);
sigaT1=transpose(siga1); % for easy viewing/debugging


%constants
Aa=3.025e+5;
alpha = -5.039;
Ea =9.891e-20;
kBoltz=1.3806e-23;

% Abscissa setup
% deltaTj=min(siga):min(siga):100; % for original temp profile
deltaTj=min(siga1):min(siga1):100; % for attenuated temp profile

Tjm=[333 353 373];%mean(sequence)+273;
Nf1=zeros([1 length(deltaTj)]);
Nf2=Nf1;
Nf3=Nf1;
i=1;
while i<length(deltaTj)
    Nf1(i)= Aa * deltaTj(i)^alpha * exp(Ea./(kBoltz * Tjm(1))); % ...
        % cycles to failure curve
    Nf2(i)= Aa * deltaTj(i)^alpha * exp(Ea./(kBoltz * Tjm(2)));
    Nf3(i)= Aa * deltaTj(i)^alpha * exp(Ea./(kBoltz * Tjm(3)));
    i=i+1;
end

%% plots single lifetime curve
figure('units','normalized','position',[.25 .25 .5 .6])
loglog(Nf3)
    h=title('IGBT Lifetime Over \DeltaT_{j} for T_{jm} = 100^{o}C');
        h.FontSize=16;
    h=ylabel('Cycles to Failure');
        h.FontSize=16;
    h=xlabel('\DeltaT_{j} (^{o}C)');
        h.FontSize=16;
    grid on

%% plots multiple lifetime curves
figure('units','normalized','position',[.25 .25 .5 .6])
loglog(deltaTj, Nf1, deltaTj, Nf2, deltaTj, Nf3)
    h=title('IGBT Lifetime Over \DeltaT_{j} for Various T_{jm}');
        h.FontSize=16;
    h=ylabel('Cycles to Failure');
        h.FontSize=16;
    h=xlabel('\DeltaT_{j} (^{o}C)');
        h.FontSize=16;
    h=legend('T_{jm} = 60^oC', 'T_{jm} = 80^oC', 'T_{jm} = 100^oC');
        h.FontSize=13;
    grid on


%% Calculate lifetime with Miner's
% % original temp profile
% sigaInt=siga*1/min(siga); % 1/min(siga) allows for '.05' in...
%     % 'siga' to correspond to index 1 of 'Nf'
% sigaInt=round(sigaInt);
% damg1=(siga./Nf1(sigaInt));
% damg2=(siga./Nf2(sigaInt));
% damg3=(siga./Nf3(sigaInt));
% damgSum1=sum(damg1);
% damgSum2=sum(damg2);
% damgSum3=sum(damg3);
% lifetime=[1/damgSum1 1/damgSum2 1/damgSum3];
% disp(['Cycles to Failure: ' num2str(lifetime)]) % this seems to...
%     % output an oddly low cycle count...?


% attenuated temp profile
sigaInt1=siga1*1/min(siga1); % 1/min(siga1) allows for '.05' in...
    % 'siga' to correspond to index 1 of 'Nf'
sigaInt1=round(sigaInt1);
damg1=siga1./Nf1(sigaInt1);
damg2=(siga1./Nf2(sigaInt1));
damg3=(siga1./Nf3(sigaInt1));
damgSum1=sum(damg1);
damgSum2=sum(damg2);
damgSum3=sum(damg3);
lifetime=[1/damgSum1 1/damgSum2 1/damgSum3];
disp(['Cycles to Failure: ' num2str(lifetime)]) % this seems to...
    % output an oddly low cycle count...?


%% Plotting lifetimes vs filter order for different Tjm
life1=[883710   261173  87966;...
    1109605 327935  110453;...
    1669031 493269  166139;...
    2789718 824479  277696;...
    2916070 861822  290273;...
    3099950 916166  308577;...
    3470437 1025661 345456;...
    4283837 1266055 426424];

%% single plot for Tjm=100
x=linspace(1,8,8);
figure('units','normalized','position',[.25 .25 .5 .6])
plot(life1(:,3));
    h=title('Lifetime vs. Filter Order for T_{jm} = 100^{o}C');
        h.FontSize=16;
    h=ylabel('Lifetime (years)');
        h.FontSize=16;
    h=xlabel('Filter Order');
        h.FontSize=16;
    xlabels = {'[Original]', 8, 16, 32, 64, 128, 256, 512};
    set(gca, 'Xtick', x, 'XtickLabel', xlabels)

%% multiplot for several Tjm's
x=linspace(1,8, 8);
figure('units','normalized','position',[.25 .25 .5 .6])
plot(x, life1(:,1), x, life1(:,2), x, life1(:,3));
    h=title('Lifetime vs. Filter Order for Differing T_{jm}');
        h.FontSize=16;
    h=ylabel('Lifetime (years)');
        h.FontSize=16;
    h=xlabel('Filter Order');
        h.FontSize=16;
    h=legend('T_{jm} = 60^{o}C', 'T_{jm} = 80^{o}C',...
        'T_{jm} = 100^{o}C');
        h.FontSize=13;
    xlabels = {'[Original]', 8, 16, 32, 64, 128, 256, 512};
    set(gca, 'Xtick', x, 'XtickLabel', xlabels)


%% ------------------ Accounting for moving Tjm  and only ambient temps
----------------------
%constants
Aa=3.025e+5;
alpha = -5.039;
Ea =9.891e-20;
kBoltz=1.3806e-23;

% %round Tj less than .001 to .001; for original
% i=1;
% while i<length(siga)+1
%     if siga(i)<.001
%         siga(i)=.001;
%     else
%     end
%     i=i+1;
% end

%round Tj less than .001 to .001; for attenuated
i=1;
while i<length(siga1)+1
    if siga1(i)<.001
        siga1(i)=.001;
    else
    end
    i=i+1;
end

% Abscissa setup
% deltaTj1=min(siga):min(siga):100; % for original temp profile
deltaTj1=min(siga1):min(siga1):100; % for attenuated temp profile

% Tjm1=rf(2,:); % extract Tjm for each rainflow cycle, original
Tjm1=rf1(2,:); % extract Tjm for each rainflow cycle, attenuated

temp=Tjm1;
Tjm1=100-max(Tjm1)+Tjm1; %temporarily shift Tjm to 100C
```

```
%% Plot old and new (shifted up to 100) Tjm
x=1:1:length(Tjm1);
figure('units','normalized','position',[.1 .25 .8 .6])
plot(x, temp, x, Tjm1)
    h=title('Mean Junction Temperatures');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Index');
        h.FontSize=16;
    h=legend('Original', 'Shifted to 100^{o}C Max');
        h.FontSize=13;

%% Create cycles to failure curves for different Tjm's
% Nfnew=zeros([length(siga) length(deltaTj1)]); % original
Nfnew=zeros([length(siga1) length(deltaTj1)]); % attenuated

i=1;
j=1;
while i<length(Tjm1)+1 % get cycles to failure curves
    while j<length(deltaTj1)
        Nfnew(i,j)= Aa * deltaTj1(j)^alpha * exp(Ea./(kBoltz *
(273+Tjm1(i))));% cycles to failure curve
        j=j+1;
    end
    i=i+1;
    j=1;
end



%%
% Miner's
% sigaInt1= siga * 1/min(siga); % original, 1/min(siga1) allows for...
    % '.05' in 'siga' to correspond to index 1 of 'Nf'
sigaInt1= siga1 * 1/min(siga1); % attenuated, 1/min(siga1) allows...
    % for '.05' in 'siga' to correspond to index 1 of 'Nf'
sigaInt1=round(sigaInt1);
i=1;

% % for original temp profile:
% while i<length(siga)+1
%     damg1(i)=siga(i)/Nfnew(i,sigaInt1(i));
%     i=i+1;
% end

% for attenuated temp profile:
while i<length(siga1)+1
    damg1(i)=siga1(i)/Nfnew(i,sigaInt1(i));
    i=i+1;
end

inv=transpose(damg1); %for debugging

damgSum1=sum(damg1);
lifetime=1/damgSum1;
```

```
disp(['Cycles to Failure: ' num2str(lifetime)])



%% Plotting lifetime vs. filter order for cycle-specific Tjm
life1 = [185140.6; 236123.7;...
    375146.3; 648668.1;...
    867113.3; 944919.6;...
    1210834.2; 1589948.9];

x=linspace(1,8, 8);
figure('units','normalized','position',[.25 .25 .5 .6])
plot(x, life1);
    h=title('Lifetime vs. Filter Order for Cycle-Specific T_{jm}');
        h.FontSize=16;
    h=ylabel('Lifetime (years)');
        h.FontSize=16;
    h=xlabel('Filter Order');
        h.FontSize=16;
    xlabels = {'[Original]', 8, 16, 32, 64, 128, 256, 512};
    set(gca, 'Xtick', x, 'XtickLabel', xlabels)

%% ----------Load and irradiance compensated lifetimes----------
% --------------------------- Case 1: max power when the irradiance is
% highest and the load demand is highest ------------------------------

%% full irradiance plot------------------------------------------------

figure('units','normalized','position',[.1 .25 .8 .6])
plot(hourlyIrradiance)
    h=title('Hourly Global Horizontal Irradiance (GHI) Data,
Albuquerque, 2014');
        h.FontSize=16;
    h=ylabel('Normalized Irradiance');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;

%% zoomed in irradiance
figure('units','normalized','position',[.25 .25 .5 .5])
plot(z(minX:maxX), hourlyIrradiance(minX:maxX))
    h=title('Hourly Global Horizontal Irradiance (GHI) Data,
Albuquerque, 2014');
        h.FontSize=16;
    h=ylabel('Normalized Irradiance');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    axis([minX,maxX,0,1]);

%% irradiance together
figure('units','normalized','position',[.1 .15 .8 .75])
plot(z, hourlyIrradiance);%, z, outhi(padLead:padTail));
grid off;
axis([1,length(hourlyIrradiance),0,1]);
ylabel('Normalized Irradiance');
```

```
title('Hourly Global Horizontal Irradiance (GHI) Data, Albuquerque,
2014');
set(gca, 'fontsize', 16, 'linewidth', 1);
% ylim([-10 2]);

props.axes1.position = [0.1 0.55 0.85 0.35];
props.axes2.position = [0.1 0.1 0.85 0.35];
props.axes1.fontsize = 16;
props.axes2.fontsize = 16;
props.axes2.linewidth = 2;
props.xlabel = 'Hour';
props.ylabel = 'Normalized Irradiance';

[ax1, ax2] = popout(gcf, 1000, 1400, props);
% h=legend('Original', 'Attenuated'); % move legend as appropriate
%     h.FontSize=13;
%     h.LineWidth=.5;


%% full load plot--------------------------------------------------

figure('units','normalized','position',[.1 .25 .8 .6])
plot(hourlyLoad)
    h=title('Typical Load Demand Profile');
        h.FontSize=16;
    h=ylabel('Normalized Power');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;

%% zoomed in load
figure('units','normalized','position',[.25 .25 .5 .5])
plot(z(minX:maxX), hourlyLoad(minX:maxX))
    h=title('Typical Load Demand Profile');
        h.FontSize=16;
    h=ylabel('Normalized Power');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    axis([minX,maxX,.4,1]);


%% load together
figure('units','normalized','position',[.1 .15 .8 .75])
plot(z, hourlyLoad);
grid off;
axis([1,length(hourlyLoad),0,1]);
ylabel('Normalized Power');
title('Typical Load Demand Profile');
set(gca, 'fontsize', 16, 'linewidth', 1);
% ylim([-10 2]);

props.axes1.position = [0.1 0.55 0.85 0.35];
props.axes2.position = [0.1 0.1 0.85 0.35];
props.axes1.fontsize = 16;
props.axes2.fontsize = 16;
props.axes2.linewidth = 2;
% props.axes2.YLim = [0 1];
props.xlabel = 'Hour';
props.ylabel = 'Normalized Power';


[ax1, ax2, rect] = popout(gcf, 1000, 1400, props);
ax1.YLim=[.25 1];
ax2.YLim=[.4 1];
rect.Position=[ax2.XLim(1) ax2.YLim(1) ax2.XLim(2)-ax2.XLim(1)...
    ax2.YLim(2)-ax2.YLim(1)];

[xf1, yf1] = popout_getcoords(ax1, 1000, 1);
[xf2, yf2] = popout_getcoords(ax2, 1000, 1);
annotation('line',[xf1 xf2],[yf1 yf2]);

[xf1, yf1] = popout_getcoords(ax1, 1400, 1);
[xf2, yf2] = popout_getcoords(ax2, 1400, 1);
annotation('line',[xf1 xf2],[yf1 yf2]);



%% full POut plot--------------------------------------------------

figure('units','normalized','position',[.1 .25 .8 .6])
plot(hourlyPOut)
    h=title('Inverter Normalized Power Output');
        h.FontSize=16;
    h=ylabel('Normalized Power');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;

%% zoomed in POut
figure('units','normalized','position',[.25 .25 .5 .5])
plot(z(minX:maxX), hourlyPOut(minX:maxX))
    h=title('Inverter Normalized Power Output');
        h.FontSize=16;
    h=ylabel('Normalized Power');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    axis([minX,maxX,0,.5]);

%% POut together
figure('units','normalized','position',[.1 .15 .8 .75])
plot(z, hourlyPOut);
grid off;
axis([1,length(hourlyPOut),0,1]);
ylabel('Normalized Power');
title('Inverter Normalized Power Output');
set(gca, 'fontsize', 16, 'linewidth', 1);
% ylim([-10 2]);

props.axes1.position = [0.1 0.55 0.85 0.35];
props.axes2.position = [0.1 0.1 0.85 0.35];
props.axes1.fontsize = 16;
props.axes2.fontsize = 16;
props.axes2.linewidth = 2;
% props.axes2.YLim = [0 1];
props.xlabel = 'Hour';
props.ylabel = 'Normalized Power';


[ax1, ax2, rect] = popout(gcf, 1000, 1400, props);
ax1.YLim=[0 .8];
ax2.YLim=[0 .6];
rect.Position=[ax2.XLim(1) ax2.YLim(1) ax2.XLim(2)-ax2.XLim(1)...
    ax2.YLim(2)-ax2.YLim(1)];

[xf1, yf1] = popout_getcoords(ax1, 1000, .6);
[xf2, yf2] = popout_getcoords(ax2, 1000, .6);
annotation('line',[xf1 xf2],[yf1 yf2]);
%
[xf1, yf1] = popout_getcoords(ax1, 1400, .6);
[xf2, yf2] = popout_getcoords(ax2, 1400, .6);
annotation('line',[xf1 xf2],[yf1 yf2]);



%% --------------------- Setting up Ptot ---------------------
minTemp=floor(min(hourlyTempData));
maxTemp=175; %Ptot = 0 at T>=175
steps=.0001; %based on fact that temps have 4 digits after decimal
PtotTemp=minTemp:steps:maxTemp;

PtotMax=zeros(length(PtotTemp),1);
i=(25-minTemp)/steps; %sets i to 25 deg C
PtotMax(1:i)=330; % Ptot = 330 at T<25C
m=(330-170)/(25-100); %slope
inter=-m*25+330; %intercept
while i<length(PtotTemp)+1 % Ptot for T>25C
    PtotMax(i)=(i*steps+minTemp)*m+inter;
    i=i+1;
end

figure('units','normalized','position',[.25 .25 .5 .6])
plot(PtotTemp, PtotMax)
    h=title('Maximum IGBT and Diode Power Dissipation vs Case
Temperature');
        h.FontSize=16;
    h=ylabel('P_{tot} (W)');
        h.FontSize=16;
    h=xlabel('T_{c} (^{o}C)');
        h.FontSize=16;
        axis([-20,maxTemp+1,0,350]);
    grid on

%% --------------------- Tc and Tj calculations ---------------------
% thermal impedances (from datasheets)
Zch=.24;
Zha=.24;
```

```
ZjcIGBT=.45;
ZjcDiode=.92;
Zjc= ZjcIGBT*ZjcDiode/(ZjcIGBT+ZjcDiode); % parallel combination...
    % of diode and IGBT


% placeholders
Tc=zeros(8760,1);
Tj=Tc;
Tj1=Tj;

%switching losses
turnOn=.000625; %joules/switch
turnOff=.001275; %joules/switch
turnTotal=turnOn+turnOff;
switchPower=turnTotal*10000; %joules*hertz=watts


PtotCalc=PtotMax(round((hourlyTempData(1)-minTemp)/steps)); %lets...
    % case temp before start time = initial ambient temp
i=1;
while i<length(Tc)+1 % calculate Tc
%     Zch=0;%r(1)*(1-exp(-i/100));
%     Zha=0;%r(1)*(1-exp(-i/100));
    if i==1
        Tc(i)=hourlyTempData(i) + PtotCalc * hourlyPOut(i) *...
            (Zch + Zha);
    else
        Ptemp=PtotMax(round(Tc(i-1)-minTemp/steps))/2 *...
            hourlyPOut(i); %power from load
        Ptemp=Ptemp+switchPower; %add switch loss
        if Ptemp>PtotMax(round(Tc(i-1)-minTemp/steps)) %check that...
                % switch loss didn't surpass max loss
            Ptemp=PtotMax(round(Tc(i-1)-minTemp/steps)); %set to max...
                %loss if surpassed
        else
        end
        Tc(i)=hourlyTempData(i) + Ptemp * (Zch + Zha);
    end
    i=i+1;
end

% INSERT PARSING FUNCTION HERE----------------------------
%---------------------------------------------------------
r=[.0872 .1599 .2020];
tau=[.000114 .001520 .020330];
% Zjc=zeros(8760,1);


i=1;
while i<length(Tj)+1 % calculate Tj
%     Zjc(i)=r(1)*(1-exp(-i/tau(1))) + r(2)*(1-exp(-i/tau(2))) +...
%         r(3)*(1-exp(-i/tau(3)));
    Tj(i)=PtotMax(round(Tc(i)-minTemp/steps))/2 * hourlyPOut(i) +...
        Zjc * Tc(i);
%     Tj1(i)=PtotMax(round(Tc(i)-minTemp/steps)) * hourlyPOut(i) +...
%         Zjc * Tc(i);
```

```
    i=i+1;
end

max(Tj)

figure('units','normalized','position',[.25 .25 .5 .6])
plot(Tj)
    h=title('Calculated Hourly T_{j}');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    xlim=[0 length(Tj)];
    axis([0,8760,-inf,inf]);


%% Tj and ambient
zeroLine=zeros(1, length(Tj));

figure('units','normalized','position',[.1 .15 .8 .75])
plot(z, Tj, z, zeroLine, z, hourlyTempData);
    h=title('Calculated Hourly T_{j}');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    h=legend('Trimmed T_{j}', 'Zero Line', 'Ambient Temp');
        h.FontSize=13;
    xlim=[0 length(Tj)];
    axis([0,8760,-inf,inf]);


%% rainflow
temp=sig2ext(Tj);
rf=rainflow(temp);
amp=rf(1,:); % this is deltaTj
Tjm=rf(2,:);
tempAmp=amp;
tempTjm=Tjm;

%round amp less than .001 to .001
i=1;
while i<length(amp)+1
    if amp(i)<.001
        amp(i)=.001;
    else
    end
    i=i+1;
end

%round Tjm less than 10.9 to 10.9...with min(amp)=.001 (from above),
%Tjm<10.9 results in NF of infinity.
i=1;
```

```
while i<length(Tjm)+1
    if Tjm(i)<10.9
        Tjm(i)=10.9;
    else
    end
    i=i+1;
end

deltaTj=min(amp):min(amp):100; % for attenuated temp profile

Nf=zeros([length(amp) length(deltaTj)]); % original

Aa=3.025e+5;
alpha = -5.039;
Ea =9.891e-20;
kBoltz=1.3806e-23;

i=1;
j=1;
while i<length(Tjm)+1 % get cycles to failure curves
    while j<length(deltaTj) % cycles to failure curve
        Nf(i,j)= Aa * deltaTj(j)^alpha * exp(Ea./(kBoltz *
(273+Tjm(i))));
        j=j+1;
    end
    i=i+1;
    j=1;
end


%% Plot deltaTj after trimming
x=1:1:length(amp);
figure('units','normalized','position',[.25 .25 .5 .6])
plot(x, amp)
    h=title('Trimmed \DeltaT_{j} From Rainflow Analysis');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Index');
        h.FontSize=16;
    xlim=[0 length(amp)];

    axis([0,length(amp),-inf, inf]);

%% Plot Tjm before/after trimming
figure('units','normalized','position',[.25 .25 .5 .6])
plot(x, tempTjm, x, Tjm)
    h=title('T_{jm} From Rainflow Analysis');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Index');
        h.FontSize=16;
    xlim=[0 length(Tjm)];
    h=legend('Original T_{jm}', 'Trimmed T_{jm}');
        h.FontSize=13;
    axis([0,length(Tjm),-inf, inf]);
```

```
%% Miner's
ampInt= amp * 1/min(amp); %1/min(amp) allows for '.05' in 'amp'...
    % to correspond to index 1 of 'Nf'
ampInt=round(ampInt);
i=1;
clear damg
damg=zeros(1,length(amp));
while i<length(amp)+1
    damg(i)=amp(i)/Nf(i,ampInt(i));
    i=i+1;
end

inv=transpose(damg); %for debugging

damgSum=sum(damg);
lifetime=1/damgSum;
disp(['Cycles to Failure: ' num2str(lifetime)])


%% ------------------------------ Case 2 ------------------------------
% -----------max load when the irradiance is above 50%-----------------
% doubles original normalized irradiance data
newHourlyIrr=hourlyIrradiance*2;

i=1;
% clips irradiance to high values of 1
% this is equivalent to full power when irr>50%
while i<length(newHourlyIrr)+1
    if newHourlyIrr(i)>1
        newHourlyIrr(i)=1;
    else
    end
    i=i+1;
end

%% full irradiance plot------------------------------------------------

figure('units','normalized','position',[.1 .25 .8 .6])
plot(newHourlyIrr)
    h=title('Doubled and Clipped GHI Data');
        h.FontSize=16;
    h=ylabel('Normalized Irradiance');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    axis([0,length(newHourlyIrr),0,1.1]);

%% zoomed in irradiance
figure('units','normalized','position',[.25 .25 .5 .5])
plot(z(minX:maxX), newHourlyIrr(minX:maxX))
    h=title('Doubled and Clipped GHI Data');
        h.FontSize=16;
    h=ylabel('Normalized Irradiance');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    axis([minX,maxX,0,1.1]);

%% irradiance together
figure('units','normalized','position',[.1 .15 .8 .75])
plot(z, newHourlyIrr);%, z, outhi(padLead:padTail));
grid off;
axis([1,length(newHourlyIrr),0,1]);
ylabel('Normalized Irradiance');
title('Doubled and Clipped GHI Data');
set(gca, 'fontsize', 16, 'linewidth', 1);
% ylim([-10 2]);

props.axes1.position = [0.1 0.55 0.85 0.35];
props.axes2.position = [0.1 0.1 0.85 0.35];
props.axes1.fontsize = 16;
props.axes2.fontsize = 16;
props.axes2.linewidth = 2;
props.xlabel = 'Hour';
props.ylabel = 'Normalized Irradiance';

[ax1, ax2] = popout(gcf, 3000, 3400, props);
% h=legend('Original', 'Attenuated'); % move legend as appropriate
%     h.FontSize=13;
%     h.LineWidth=.5;



%% hourly POut
newHourlyPO = newHourlyIrr.*hourlyLoad; % new irradiance-load product



%% full POut plot--------------------------------------------------

figure('units','normalized','position',[.1 .25 .8 .6])
plot(newHourlyPO)
    h=title('Inverter Normalized Power Output');
        h.FontSize=16;
    h=ylabel('Normalized Power');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;

%% zoomed in POut
figure('units','normalized','position',[.25 .25 .5 .5])
plot(z(minX:maxX), newHourlyPO(minX:maxX))
    h=title('Inverter Normalized Power Output');
        h.FontSize=16;
    h=ylabel('Normalized Power');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    axis([minX,maxX,0,1]);
```

```
%% POut together
figure('units','normalized','position',[.1 .15 .8 .75])
plot(z, newHourlyPO);
grid off;
axis([1,length(hourlyPOut),0,1]);
ylabel('Normalized Power');
title('Inverter Normalized Power Output');
set(gca, 'fontsize', 16, 'linewidth', 1);
% ylim([-10 2]);

props.axes1.position = [0.1 0.55 0.85 0.35];
props.axes2.position = [0.1 0.1 0.85 0.35];
props.axes1.fontsize = 16;
props.axes2.fontsize = 16;
props.axes2.linewidth = 2;
% props.axes2.YLim = [0 1];
props.xlabel = 'Hour';
props.ylabel = 'Normalized Power';

[ax1, ax2, rect] = popout(gcf, 1000, 1400, props);

%% --------------------- Tc and Tj calculations ---------------------
% thermal impedances (from datasheets)
Zch=.24;
Zha=.24;
ZjcIGBT=.45;
ZjcDiode=.92;
Zjc= ZjcIGBT*ZjcDiode/(ZjcIGBT+ZjcDiode); % parallel combination...
    % of diode and IGBT


% placeholders
Tc=zeros(8760,1);
Tj=Tc;
Tj1=Tj;

%switching losses
turnOn=.000625; %joules/switch
turnOff=.001275; %joules/switch
turnTotal=turnOn+turnOff;
switchPower=turnTotal*10000; %joules*hertz=watts


PtotCalc=PtotMax(round((hourlyTempData(1)-minTemp)/steps)); %lets...
    % case temp before start time = initial ambient temp
i=1;
while i<length(Tc)+1 % calculate Tc
%     Zch=0;%r(1)*(1-exp(-i/100));
%     Zha=0;%r(1)*(1-exp(-i/100));
    if i==1
        Tc(i)=hourlyTempData(i) + PtotCalc * newHourlyPO(i) *...
            (Zch + Zha);
    else
        Ptemp=PtotMax(round(Tc(i-1)-minTemp/steps))/2 *...
            newHourlyPO(i); %power from load
        Ptemp=Ptemp+switchPower; %add switch loss
        if Ptemp>PtotMax(round(Tc(i-1)-minTemp/steps)) %check that...
```

```
                % switch loss didn't surpass max loss
            Ptemp=PtotMax(round(Tc(i-1)-minTemp/steps)); %set to max...
                %loss if surpassed
        else
        end
        Tc(i)=hourlyTempData(i) + Ptemp * (Zch + Zha);
    end
    i=i+1;
end

% INSERT PARSING FUNCTION HERE----------------------------
%---------------------------------------------------------
r=[.0872 .1599 .2020];
tau=[.000114 .001520 .020330];
% Zjc=zeros(8760,1);


i=1;
while i<length(Tj)+1 % calculate Tj
%     Zjc(i)=r(1)*(1-exp(-i/tau(1))) + r(2)*(1-exp(-i/tau(2))) +...
%         r(3)*(1-exp(-i/tau(3)));
    Tj(i)=PtotMax(round(Tc(i)-minTemp/steps))/2 * newHourlyPO(i) +...
        Zjc * Tc(i);
%     Tj1(i)=PtotMax(round(Tc(i)-minTemp/steps)) * hourlyPOut(i) +...
%         Zjc * Tc(i);

    i=i+1;
end

max(Tj)

figure('units','normalized','position',[.25 .25 .5 .6])
plot(Tj)
    h=title('Calculated Hourly T_{j}');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    xlim=[0 length(Tj)];
    axis([0,8760,-inf,inf]);


%% Tj and ambient
zeroLine=zeros(1, length(Tj));

figure('units','normalized','position',[.1 .15 .8 .75])
plot(z, Tj, z, zeroLine, z, hourlyTempData);
    h=title('Calculated Hourly T_{j}');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Hour');
        h.FontSize=16;
    h=legend('Trimmed T_{j}', 'Zero Line', 'Ambient Temp');
        h.FontSize=13;
    xlim=[0 length(Tj)];
    axis([0,8760,-inf,inf]);


%% rainflow
temp=sig2ext(Tj);
rf=rainflow(temp);
amp=rf(1,:); % this is deltaTj
Tjm=rf(2,:);
tempAmp=amp;
tempTjm=Tjm;

%round amp less than .001 to .001
i=1;
while i<length(amp)+1
    if amp(i)<.001
        amp(i)=.001;
    else
    end
    i=i+1;
end

%round Tjm less than 10.9 to 10.9...with min(amp)=.001 (from above),
%Tjm<10.9 results in NF of infinity.
i=1;
while i<length(Tjm)+1
    if Tjm(i)<10.9
        Tjm(i)=10.9;
    else
    end
    i=i+1;
end

deltaTj=min(amp):min(amp):100; % for attenuated temp profile

Nf=zeros([length(amp) length(deltaTj)]); % original

Aa=3.025e+5;
alpha = -5.039;
Ea =9.891e-20;
kBoltz=1.3806e-23;

i=1;
j=1;
while i<length(Tjm)+1 % get cycles to failure curves
    while j<length(deltaTj) % cycles to failure curve
        Nf(i,j)= Aa * deltaTj(j)^alpha * exp(Ea./(kBoltz *
(273+Tjm(i))));
        j=j+1;
    end
    i=i+1;
    j=1;
end


%% Plot deltaTj after trimming
```

```
x=1:1:length(amp);
figure('units','normalized','position',[.25 .25 .5 .6])
plot(x, amp)
    h=title('Trimmed \DeltaT_{j} From Rainflow Analysis');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Index');
        h.FontSize=16;
    xlim=[0 length(amp)];

    axis([0,length(amp),-inf, inf]);

%% Plot Tjm before/after trimming
figure('units','normalized','position',[.25 .25 .5 .6])
plot(x, tempTjm, x, Tjm)
    h=title('T_{jm} From Rainflow Analysis');
        h.FontSize=16;
    h=ylabel('Temperature (^{o}C)');
        h.FontSize=16;
    h=xlabel('Index');
        h.FontSize=16;
    xlim=[0 length(Tjm)];
    h=legend('Original T_{jm}', 'Trimmed T_{jm}');
        h.FontSize=13;
    axis([0,length(Tjm),-inf, inf]);


%% Miner's
ampInt= amp * 1/min(amp); %1/min(amp) allows for '.05' in 'amp'...
    % to correspond to index 1 of 'Nf'
ampInt=round(ampInt);
i=1;
clear damg
damg=zeros(1,length(amp));
while i<length(amp)+1
    damg(i)=amp(i)/Nf(i,ampInt(i));
    i=i+1;
end

inv=transpose(damg); %for debugging

damgSum=sum(damg);
lifetime=1/damgSum;
disp(['Cycles to Failure: ' num2str(lifetime)])
```

REFERENCES

[1] Darling, Seth B., Fengqi You, Thomas Veselka, and Alfonso Velosa. "Assumptions and the Levelized Cost of Energy for Photovoltaics." *Energy & Environmental Science Energy Environ. Sci.* 4.9 (2011): 3133. Web.

[2] GopiReddy, L., L. Tolbert, B. Ozpineci, and J. Pinto. "Rainflow Algorithm-Based Lifetime Estimation of Power Semiconductors in Utility Applications." *IEEE Transactions on Industry Applications* 51.4 (2015): 3368-375. *IEEE Explore*. Web. 27 Mar. 2016.

[3] Anurag, Anup, Yongheng Yang, and Frede Blaabjerg. "Impact of Reactive Power Injection outside Feed-in Hours on the Reliability of Photovoltaic Inverters." *2015 IEEE 6th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)* (2015): n. pag. *IEEE Xplore*. Web. 27 Mar. 2016.

[4] Berg, H., and E. Wolfgang. "Advanced IGBT Modules for Railway Traction Applications: Reliability Testing." *Microelectronics Reliability* 38.6-8 (1998): 1319-323. *ScienceDirect*. Web. 27 Mar. 2016.

[5] Busca, Cristian. "Modeling Lifetime of High Power IGBTs in Wind Power Applications - An Overview." *2011 IEEE International Symposium on Industrial Electronics* (2011): n. pag. *IEEE*. Web. 27 Mar. 2016.

[6] MacKenzie, Harland. "Accelerated Testing with the Inverse Power Law." *Reliability Edge* 12 (2012): 1, 20-21, 23-25. Print.

[7] Blake, Carl, and Chris Bull. *IGBT or MOSFET: Choose Wisely*. Rep. International Retifier, n.d. Web. 27 Mar. 2016.

[8] Schulz, Martin. *Thermal Interface A Key Factor in Improving Lifetime in Power Electronics*. Tech. Infineon Technologies, n.d. Web. 27 Mar. 2016.

[9] Held, M., P. Jacob, G. Nicoletti, P. Scacco, and M.-H. Poech. "Fast Power Cycling Test of IGBT Modules in Traction Application." *Proceedings of Second International Conference on Power Electronics and Drive Systems* 1 (1997): 425-30. *IEEE*. Web. 27 Mar. 2016.

[10] Bayerer, R., T. Herrmann, T. Licht, J. Lutz, and M. Feller. "Model for Power Cycling Lifetime of IGBT Modules - Various Factors Influencing Lifetime." (2008): 1-6. *IEEE*. Web. 27 Mar. 2016.

[11] Vogel, Klaus, Alexander Ciliox, and Andreas Schmal. *IGBT with Higher Operation Temperature - Power Density, Lifetime and Impact on Inverter Design*. Tech. Infineon Technologies, n.d. Web. 27 Mar. 2016.

[12] *Thermal Analysis of Semiconductor Systems*. White Paper. Freescale Semiconductors, n.d. Web. 27 Mar. 2016.

[13] Schutze, Thomas. *Thermal Equivalent Circuit Models*. Application Note. Infineon Technologies, 16 June 2008. Web. 27 Mar. 2016.

[14] Nagode, M., and M. Hack. "An Online Algorithm for Temperature Influenced Fatigue Life Estimation: Stress–life Approach." *International Journal of Fatigue* 26.2 (2003): 163-71. *ScienceDirect*. Web. 27 Mar. 2016.

[15] Waegter, John. *Note 5.1 Stress Range Histories and Rain Flow Counting*. June 2009.

[16] "The NSRDB Data Viewer." *The NSRDB Data Viewer*. National Renewable Energy Laboratories, n.d. Web. 27 Mar. 2016.

[17] Nieslony, Adam. *Rainflow Counting Algorithm*. *File Exchange*. Vers. 1.2. MathWorks, n.d. Web. 27 Mar. 2016.

[18] "National Grid - Load Profiles." *National Grid - Load Profiles*. NationalGridUS, n.d. Web. 27 Mar. 2016.

[19] Sawle, Andrew, and Arthur Woodworth. *Mounting Guidelines for the Super-247 (TM)*. Application Note AN-997. International Rectifier, n.d. Web. 27 Mar. 2016.

[20] Schnell, R., M. Bayer, and S. Geissmann. Thermal Design and Temperature Ratings of IGBT Modules. Application Note 5SYA 2093-00. ABB Switzerland Ltd. Semiconductors, n.d. Web. 27 Mar. 2016.

[21] Romeu, Jorge Luis. "Understanding Series and Parallel Systems Reliability." *START - Selected Topics in Assurance Related Technologies* 11 (2004-2005): 1-8. Alion System Reliability Center. Web. 27 Mar. 2016.

[22] Kovacevic, I. F., U. Drofenik, and J. W. Kolar. "New Physical Model for Lifetime Estimation of Power Modules." *The 2010 International Power Electronics Conference - ECCE ASIA* - (2010): n. pag. Web. 27 Mar. 2016.

[23] Scheuermann, U., R. Schmidt, and P. Newman. "Power Cycling Testing with Different Load Pulse Durations." *7th IET International Conference on Power Electronics, Machines and Drives (PEMD 2014)* (2014): n. pag. Web. 27 Mar. 2016.