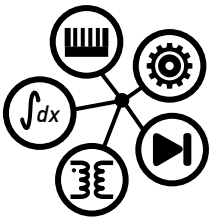

PLECS

DEMO MODEL

Rainflow Counting and Lifetime Prediction

For a drive with direct torque control using PLECS simulation scripts

Last updated in PLECS 4.6.1

1 Overview

This model features a rainflow counting algorithm that predicts the lifetime of semiconductor devices in a direct torque-controlled (DTC) motor drive under different load conditions. The calculation is based on the temperature profile of the power semiconductors from a transient electro-thermal simulation.

2 Model

The system overview is shown in Fig. 1. This demo model does not focus on the DTC model itself but rather on the rainflow counting algorithm to predict the semiconductor lifetime. The DTC application itself is described in the demo model “Induction Machine Drive Controlled with DTC”. The demo model

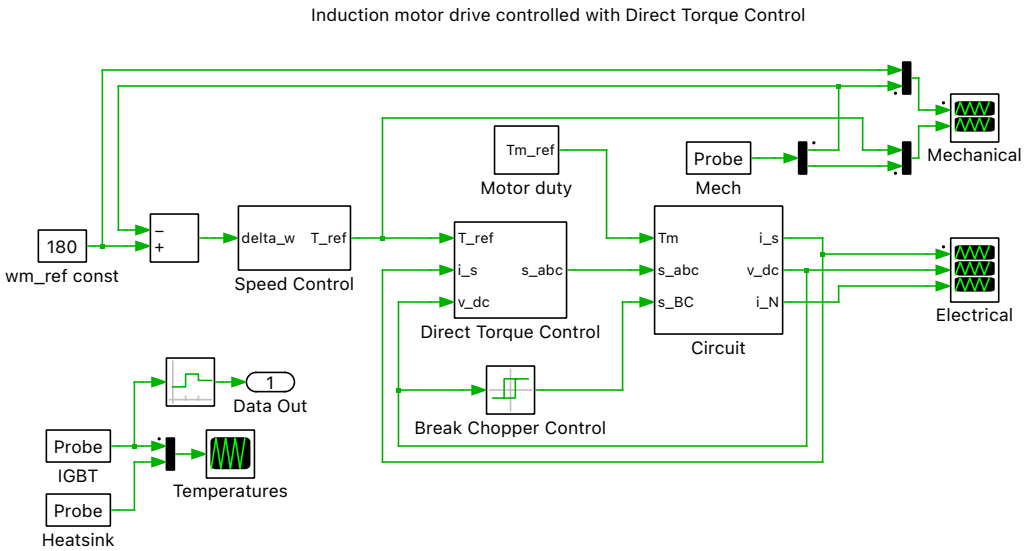

Figure 1: System overview

uses a simulation script for performing the lifetime analysis. To access the simulation results inside the simulation script a Signal Outport is placed on the root schematic (named “Data Out”). The IGBT junction temperatures and the simulation time are then returned from the transient simulation to the the simulation script. To limit the analysis of peaks and valleys in the temperature profile, the small ripple due to the converter switching operation is filtered with a Periodic Average block and averaging time of 1 ms. The lifetime calculation is dominated by larger temperature swings and not small switching ripples.

2.1 Simulation Script

The simulation script can be opened from the menu **Simulation + Simulation Scripts** in PLECS Standalone and from the corresponding m-file `rainflow_counting_script.m` in the PLECS Blockset version. There is also a Python script that implements the same lifetime calculation using the XML-RPC interface of PLECS Standalone.

There are several files packaged with this demo:

- `find_peaks.m`: Script that calculates all extreme points (valleys and peaks) in the simulated junction temperature profile
- `gen_histogram.m`: Script that generates the histogram of the calculated rainflows
- `predict_lifetime.m`: Script that estimates the lifetime of the IGBT module
- `rainflow_algorithm.m`: Script that implements the actual rainflow counting algorithm

- `rainflow_counting_plecs`: Folder that contains the thermal description files for the transient electro-thermal simulation
- *Only for PLECS Standalone* – `python`: Folder that contains the Python sources for this demo model
- *Only for PLECS Blockset* – `rainflow_counting_script.m`: The overall simulation script that runs the PLECS simulation and the individual m-files from above

Script Overview

The simulation script is structured in different parts where the actions are executed in the following order:

- Select the load profile.
- Run a transient electro-thermal simulation that calculates the junction temperature of the semiconductors $T_j(t)$.
- Find the local temperature minima and maxima and the time at which these occur.
- Run the rainflow counting algorithm on the peak and valley data. This calculates amplitude, ending point and time span of each rainflow.
- Count the number of full temperature cycles with equal amplitude.
- Predict the lifetime using the rainflow data.

Plotting

Different plots are shown during execution of the script. The first plot shows that actual peaks and valleys in the junction temperature results. The information of `cyc_ampl_s[]` and `cyc_freq_s[]` is shown in a histogram with a specific bin width (default bin width is $5\,^{\circ}\mathrm{C}$).

3 Rainflow Counting Algorithm

The developed algorithm is based on the ASTM standard practices for cycle counting in fatigue analysis. The rainflow counting principle has four rules:

1 A rainflow is started at each peak and valley;

2 For a flow started at a peak, it is stopped by a subsequent higher peak;

3 For a flow started at a valley, it is stopped by a subsequent lower valley;

4 A rainflow is stopped when it encounters a previous rainflow. This rule is tricky and implies two conditions:

  1 The previous rainflow must be started at no lower (higher than or equal) a peak or no higher (deeper than or equal) a valley (otherwise rule 2 or 3 described above stops the previous rainflow)

  2 The two rainflows can meet (depending on the structure of roofs).

To determine the stop of a rainflow which starts at a peak, the height of the then-current peak (P_{cur}) is compared to its previous peaks (rule 4 condition 1)) and its subsequent peaks (rule 2). Note that if there exist multiple previous peaks higher than or equal to P_{cur}, only the peak ($P_{\mathrm{pre,h,n}}$) nearest to P_{cur} needs to be considered. The rainflow from $P_{\mathrm{pre,h,n}}$ may be intercepted by an earlier rainflow before P_{cur}, but the earlier rainflow will also flow through the remaining roof(s) and intercept the rainflow from P_{cur} at the same place. For the subsequent peaks, only the higher peak ($P_{\mathrm{sub,h,n}}$) nearest to P_{cur} needs to be considered. And, if the lowest valley ($V_{\mathrm{pre,l}}$) between $P_{\mathrm{pre,h,n}}$ and P_{cur} is no lower than the lowest valley ($V_{\mathrm{sub,l}}$) between $P_{\mathrm{sub,h,n}}$ and P_{cur}, then rule 4 condition 2) is satisfied. Otherwise, the rainflow from P_{cur} is stopped before $P_{\mathrm{sub,h,n}}$. The rainflows that start at a valley follow the same logic and are not presented here explicitly.

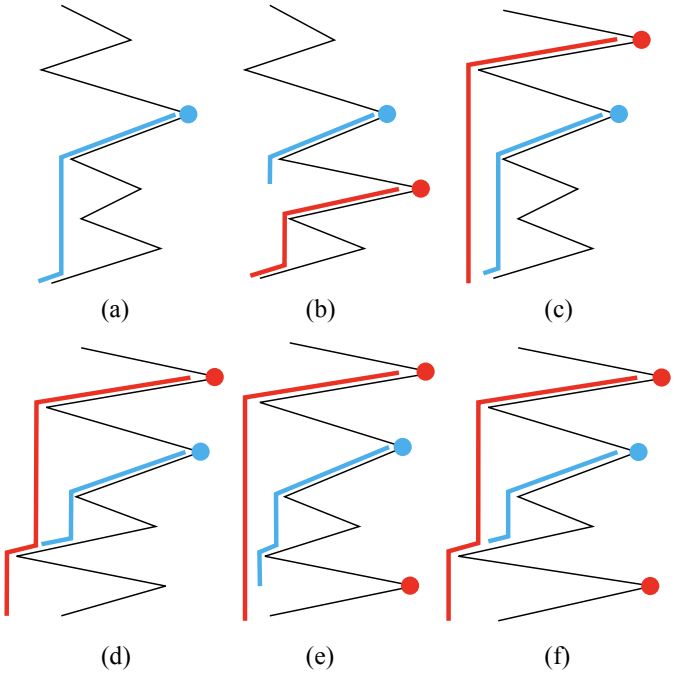

Figure 2: Illustration of rainflow started at a current peak (marked by the blue dot): (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, and (f) case 6

In the developed algorithm, the rainflow counting rules result in 6 cases for rainflows started at peaks and 6 cases for rainflows started at valleys. The description and figure below demonstrate the 6 different cases that need to be considered. P_{cur} is marked by a blue dot and $P_{\mathrm{pre,h,n}}$ and $P_{\mathrm{sub,h,n}}$ are marked by red dots.

1 $P_{\mathrm{pre,h,n}}$ and $P_{\mathrm{sub,h,n}}$ do not exist; the rainflow from P_{cur} stops at the end of time.

2 $P_{\mathrm{pre,h,n}}$ does not exist but $P_{\mathrm{sub,h,n}}$ exists; the rainflow from P_{cur} is stopped before $P_{\mathrm{sub,h,n}}$.

3 $P_{\mathrm{pre,h,n}}$ exists but $P_{\mathrm{sub,h,n}}$ does not exist, and rule 4 condition 2) is not satisfied; the rainflow from P_{cur} stops at the end of time.

4 $P_{\mathrm{pre,h,n}}$ exists but $P_{\mathrm{sub,h,n}}$ does not exist, and rule 4 condition 2) is satisfied; the rainflow from P_{cur} is intercepted by the previous rainflow from $P_{\mathrm{pre,h,n}}$ or an earlier peak.

5 Both $P_{\mathrm{pre,h,n}}$ and $P_{\mathrm{sub,h,n}}$ exist, and rule 4 condition 2) is not satisfied; the rainflow from P_{cur} is stopped before $P_{\mathrm{sub,h,n}}$.

6 Both $P_{\mathrm{pre,h,n}}$ and $P_{\mathrm{sub,h,n}}$ exist, and rule 4 condition 2) is satisfied; the rainflow from P_{cur} is intercepted by the pervious rainflow from $P_{\mathrm{pre,h,n}}$ or an earlier peak.

The ending points of the 6 cases are illustrated in Fig. 2. The 6 cases for a rainflow starting at a valley are determined by the same rules and are not explicitly presented here.

3.1 Implementation

For the calculation the rainflow counting algorithm uses the following data structures:

- `A[i]`: The amplitude of the rainflow that started at the ith extreme point (peak or valley point).
- `i_flow_end[i]`: The endpoint of a rainflow. The expression `i_flow_end[7] = 11`, for example, means that the rainflow which started at the 7th peak point ended at the 11th peak point.
- `t_cyc_start[i]`: The start time of the ith rainflow.
- `t_cyc_end[i]`: The stop time of the ith rainflow.

The rainflows or half cycles with equal amplitudes are counted and sorted in descending order of amplitude. The information of amplitudes and frequencies are stored in the two lists, `cyc_ampl_s[]` and `cyc_freq_s[]`.

3.2 Lifetime Estimation

The function `predict_lifetime()` predicts the lifetime of an IGBT module. The input parameters are: `a` and `b` of the lifetime model, `hists[]` and `bins[]` from `histogram()` and the time period of the load cycle. The lifetime model is

$$N_\mathrm{f} = a \cdot (\Delta T_\mathrm{j})^{-b}, \tag{1}$$

which relates the number of cycles to failure, N_f, to the amplitude of a T_j cycle, ΔT_j. The constants a and b are determined by fitting the accelerated power cycling life curve of the IGBT module. The Miner's rule is used to normalize the damage caused by cycles with different ΔT_j that appeared in the junction temperature waveform. The normalized damages are stored in the list `acc_damage[]`. The sum of these damages is the total damage under a load cycle. The IGBT module fails after a number of such load cycles when the accumulated damage reaches 1. The predicted lifetime of a device is a product of the time period of the load cycle, t_period, and the number of load cycles to failure, PC.

4 Simulation

This example demonstrates a lifetime prediction for an IGBT inverter power module in an induction motor (IM) drive operated under two load duties. The IM drive circuit and control scheme are described in the PLECS demo model "Induction Machine Drive Controlled with DTC". A thermal model for the inverter is added to simulate the junction temperature of the IGBTs, in this case the PM75CLA060 power module from POWEREX [1]. The power cycling curve of this module is provided in the Mitsubishi PV-IPM application note [2], and is shown in Fig. 3.

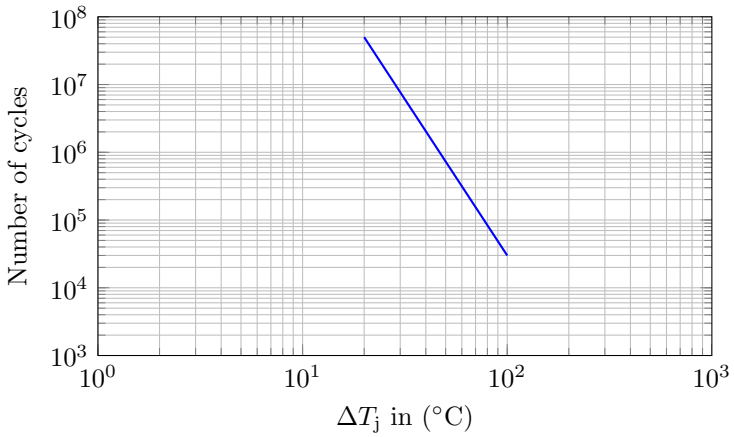

Figure 3: The power cycling data for power module PM75CLA060 from POWEREX

The IM is spinning at $180\,\mathrm{rad/s}$. The IEC 60034-1 Standard defines 10 different types of load cycles [3]. Two of them are can be simulated in this demo model:

- **S1** refers to a constant load scenario. In this example the load torque is fixed at $40\,\mathrm{Nm}$.
- **S6** refers to a continuous periodic load scenario. In this example the load torque varies periodically between $20\,\mathrm{Nm}$ and $60\,\mathrm{Nm}$ (using a 0.5 duty ratio with a period of 5 seconds).

4.1 Junction Temperatures during System Start

The motor starting current is high, and large junction temperature variations are observed. The initial temperature of the inverter module and its heat sink is the same as the ambient temperature. Due to the relatively large thermal capacitance of the heat sink, the temperature of heat sink takes about 7 seconds to reach steady state. The thermal damage in this 7 second period is mainly due to IM drive and cooling start and does not reflect the impact of duty types on inverter lifetimes.

4.2 Junction Temperature Cycles

After the initial startup transient, the motor drive and cooling system is under steady-state operation. The IGBT junction temperature variations are generated by drive operations and reflect stresses under the two load duties. The steady-state junction and heat sink temperature under the two load cycles S1 and S6 are shown in Fig. 4.

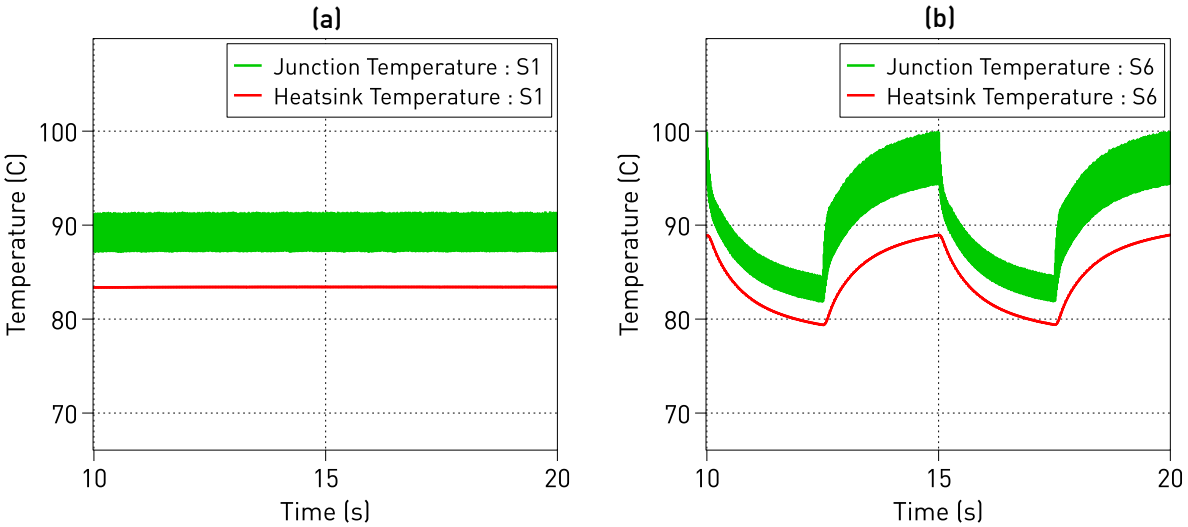

Figure 4: The steady-state junction and heat sink temperature under: (a) S1 duty, and (b) S6 duty

4.3 Rainflow Counting and Lifetime Prediction

To run the rainflow analysis open the menu **Simulation + Simulation Scripts** in PLECS Standalone or the m-file `rainflow_counting_script.m` in PLECS Blockset. With the variable `Load_profile` one can choose if scenario *S1* (`Load_profile = 1`) or *S6* (`Load_profile = 2`) should be run.

The first output of the script are the extreme points of the junction temperature data, as shown in Fig. 5. After performing the rainflow counting algorithm using the peak and valley points of the junction temperature the `generate_histogram()` function is called. It calculates how many temperature cycles fall into cycle bins of width $5\,^{\circ}\mathrm{C}$. For the load scenario S6 this leads to the results in Fig. 6. For the load scenario S1 there is only one bin size of $\Delta T_\mathrm{j} = 5\,^{\circ}\mathrm{C}$ and the number of full cycles is 46.5.

Using the histogram data the predicted lifetimes of the devices are as follows:

- If the motor is operating under S1 duty, the junction temperature cycles are all less than $5\,^{\circ}\mathrm{C}$; the predicted lifetime of the inverter under S1 duty is 16 years.
- If the motor is operating under S6 duty, each load cycle ($5\,\mathrm{s}$) results in a relatively large 15 and $20\,^{\circ}\mathrm{C}$ junction temperature cycle; the predicted lifetime of the inverter under S6 duty is 1.1 years.

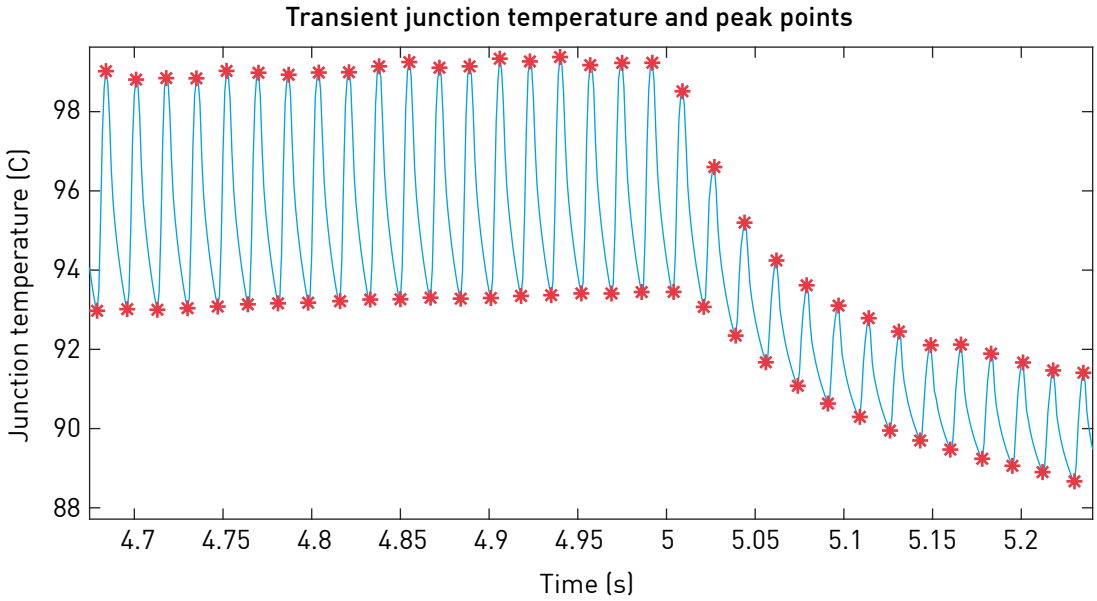

Figure 5: An excerpt of the extreme points for the load scenario S6. The junction temperature data is filtered with a periodic average filter with an averaging interval of $1\,\mathrm{ms}$.

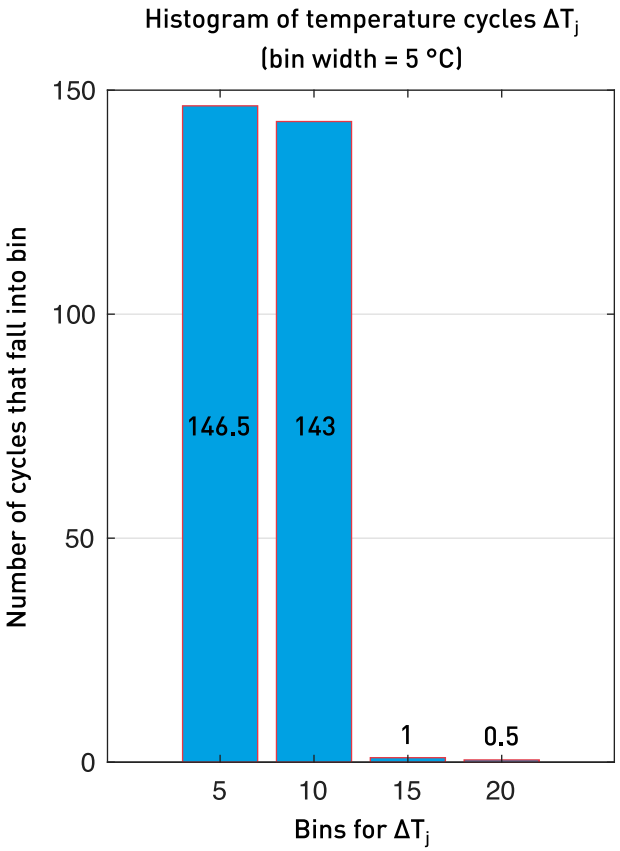

Figure 6: Histogram for junction temperature cycles

5 Conclusion

The results show that the lifetime of the IGBT module is limited by only a few large T_j cycles. The inverter has a much shorter lifetime when the motor is subject to periodic load torque changes compared

to a constant load torque.

References

[1] MITSUBISHI PM75CLA060FLAT-B Datasheet, [Online]. Available: https://www.mitsubishielectric-mesh.com/products/pdf/PM75CLA060_e.pdf. [Accessed: Aug. 3rd, 2021].

[2] PV-IPM Application Note, [Online]. Available: https://www.mitsubishielectric.com/semiconductors/files/manuals/pv_ipm_note_e2.pdf. [Accessed: Aug. 3rd, 2021].

[3] IEC 60034-1 Duty Cycles, [Online]. Available: https://avsld.com.sg/iec-duty-cycles/. [Accessed: Aug. 3rd, 2021].

Revision History:

PLECS 4.6.1	First release

How to Contact Plexim:

☎	+41 44 533 51 00	Phone
	+41 44 533 51 01	Fax
✉	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	`info@plexim.com`	Email
	`http://www.plexim.com`	Web

PLECS Demo Model