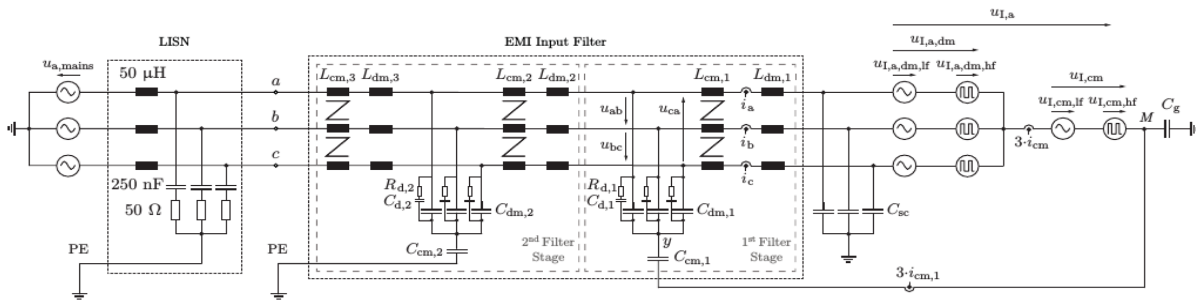
Three-Phase EMI Filter Optimizer Manual

Andrija Stupar, December 2014



About

The Three-Phase EMI Filter Optimizer has been created as a result of PhD studies and research by Andrija Stupar, while he was an “external” PhD student at the Power Electronics Systems Laboratory at ETH Zurich, in the period from April to December 2014. The aim was to create a software tool that can optimize EMI filters to three-phase converters using a database of real components, and that can therefore be useful to create practical designs for ordinary engineers. However, keep in mind that this tool, as supplied with this manual, is **NOT** a commercial tool but a **research demonstrator**. Thus, although it aims to be as stable, reliable, and accurate as possible, and can be used as a valuable aid for creating practical EMI filter designs, it is not fully geared towards every-day “ordinary” engineering use. For example, the many options on design and optimization exist in order to compare different approaches, which was useful in the afore-mentioned reserch. For that reason, the tool is useful both for researchers wanting to explore theoretical limits as well as engineers wanting to build practical optimized filters, but with limitations. Please keep this in mind and be careful when setting the many options of the program so that are selected according to your desired application.

Disclaimer

The software is provided AS IS, and the author takes NO RESPONSIBILITY OR LIABILITY WHATSOEVER for any possible outcome of any action or activity you may take or perform with it. The Three-Phase EMI Filter Optimizer is provided with NO WARRANTY OF ANY KIND; INCLUDING NO WARRANTY FOR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. By using the software you agree to these conditions.

Licensing

For licensing information, please click the “About” button at the bottom of the Optimizer's main window. You must obey the terms of the various licenses specified therein. By using the software you agree to these conditions. The vast majority of the code is owned solely by Andrija Stupar, but other portions of it are owned and co-owned by the Power Electronics Systems Laboratory of ETH Zurich and Gecko-Simulations AG. Note that the Optimizer depends on using GeckoCIRCUITS and a custom version of GeckoMAGNETICS, which are products of Gecko-Simulations AG.

Questions, Comments, Suggestions, Bugs, etc.

Please send any of the above to one of the three e-mail addresses:
stupar@lem.ee.ethz.ch
andrija.stupar@gecko-simulations.com
andrija@stupar.com
Any and all feedback is very welcome!

1. Introduction

The Three-Phase EMI Filter Optimizer is a software tool uses circuit simulation and magnetic component modelling to produce multiobjective pareto-optimal designs of EMI filters for three-phase power electronic converters. There are two optimization objectives: **efficiency (losses)** and **power density (volume)**. The tool computes the set of pareto-optimal solutions – that set of designs such that, for a given efficiency, no other design exists with the same effiency and greater power density (and vice versa) – and then picks, based on the optimization goal defined by the user, the best filter design from that pareto set.

1.1 What this tool does and will provide

This tool will determine the optimal
- distribution of attenuations per filter stage
- number of stages
- L and C component values
- construction of inductors and capacitors – which core, wire, capacitor component etc. to use to build your filter.

1.2 What this tool does NOT do and will NOT provide

This tool does NOT take into consideration the physical placement of the components, parasitic effects and couplings of the components, and complex permeabilities. Also, it does NOT perform any 3D simulations of any sort.

1.3 System Requirements

In order to run the Optimizer, you **MUST** have Java installed on your system. The minimum required version is **Java 7**. If necessary, to download the appropriate Java Runtime Environment (JRE) for your system, go to http://java.com/download/.

This tool is intended to be used on powerful computers with plenty of available RAM. The amount of RAM required depends on the particular optimization problem – the larger the design space which needs to be explored, the larger the memory requirements. You should run the program on a machine with AT LEAST 4 GB or RAM, although 8 GB of RAM are recommended. Also, multi-processor machines are highly recommended, as the program can take advantage of parallelization in that case. The ideal machine which will be able to tackle a wide variety of large design spaces is a fast server with 8 – 16 cores available and more than 10 GB of RAM. If you do not have access to such a machine, consider using a cloud computing platform, such as Google Compute Engine.

1.4 Warning

Please do not just open the program and blindly click about until you are able to start an optimization. The result may be a really large design space which will result in an optimization procedure that will consume a large amount of your computer's resources (or even outstrip them) and take many days to complete. Instead, follow the examples given in this manual for testing the tool.

1.5 Starting the Optimizer

On most systems, it should be possible to start the program by double-clicking the EMI_3phOptimizer.jar file. This, however, is **NOT** recommended, as it will likely not allocate enough memory to the application. If you wish to control how much memory is allocated to the Optimizer, start the EMI_3phOptimizer.jar from the command line, using the following command

```
java -Xmx4096m -XX:UseGCOverheadLimit -jar EMI3ph_Optimizer.jar
```

This will allocate 4 GB (4096 MB) for the Optimizer. You can change the amount allocated by changing that number above. Do not forget the second argument starting with "-XX", or the application may terminate unexpectedly when an optimization is started.

The other option is to double-click the file **StartOptimizer.jar** (or run it from a command line using (java -jar StartOptimizer.jar). This will automatically determine the amount of RAM on your system, and allocate 90% of it for the Optimizer. Using StartOptimizer.jar will work on Windows systems (Windows 7 and above, and probably some older versions as well) and Linux systems. It will NOT work on Mac OS systems. If clicking StartOptimizer.jar does nothing, start the program from the command line as explained above.

2. Setting up the Circuit Simulation Models

The basic principle of the Optimizer is simulation of circuit models of an EMI filter in the fast power electronics simulator **GeckoCIRCUITS**[1], and then the extraction of waveforms for further analysis, based on which the components themselves are designed. GeckoCIRCUITS is bundled with the Optimizer. Therefore, appropriate GeckoCIRCUITS simulation models must exist for the Optimizer to work.

2.1 Three-Phase EMI Filter Topology

The filter topology considered by the Optimizer is **fixed**. The filter considered is a classic *LCLC* structure, with **discrete** common-mode and differential-mode inductors and capacitors. Each differential-mode (DM) stage is followed by a common-mode (CM) stage, and the number of DM and CM stages **must** be equal. The only option left to the user with regards to the topology, is to select, for each stage, whether the common-mode filtering capacitors are in series with the differential-mode filtering capacitors, or whether they are in parallel. The topology[2] is shown in Figure 1.

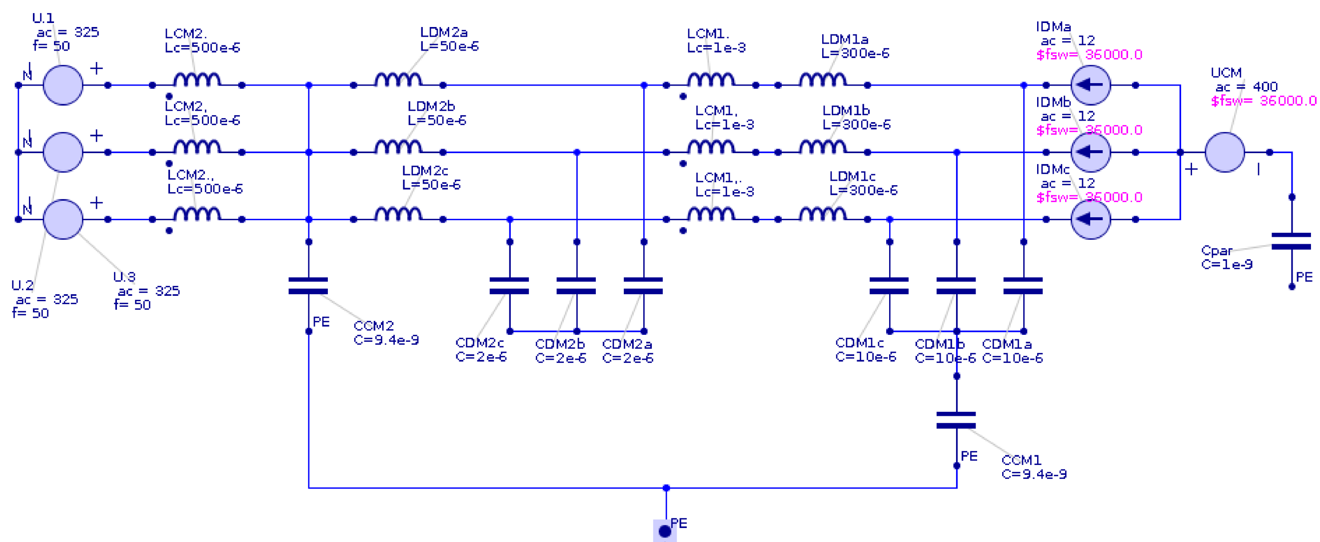


Figure 1: The EMI filter topology considered by the Optimizer. Shown is both a stage where the CM capacitance is in series with the DM capacitance, and where it is not.

1 See more at http://www.gecko-simulations.com/geckocircuits.html

2 The topology schematic is taken from D. O. Boillat, J. W. Kolar, J. Mühlethaler, "Volume Minimization of the Main DM/CM EMI Filter Stage of a Bidirectional Three-Phase Three-Level PWM Rectifier System", Proceedings of the IEEE Energy Conversion Congress and Exposition (ECCE USA 2013), Denver, Colorado, USA, September 15-19, 2013.

2.2 Differential-Mode Simulation Model

The filter topology of Figure 1 cannot be meaningfully simulated in GeckoCIRCUITS as a whole. It is necessary therefore to separately simulate a DM noise model and a CM noise model. You must construct a simulation model for both differential-mode and common-mode filters that suits your application. Sample filter and noise models are given in the "models" subdirectory of the Optimizer package. The two-stage DM filter simulation model is given in Figure 2. It is fairly universal, and contains all of the basic elements a DM filter model should have. Note the necessary components:

- all **three** phases
- for each phase and stage, the differential mode filtering components: *L* and *C*
- and inductor component in each phase and stage representing the **leakage inductance** of the common-mode inductance (choke) of that stage
- the mains voltage represented by voltage sources
- the **noise sources**: three current sources representing the differential-mode current of the converter at each phase. Rather than simulating the entire converter (which is possible, but not recommended, due to the slower simulation in that case), the noise (DM current) waveform is extracted from a single simulation of the entire converter, saved to a file, and then loaded into the current sources.

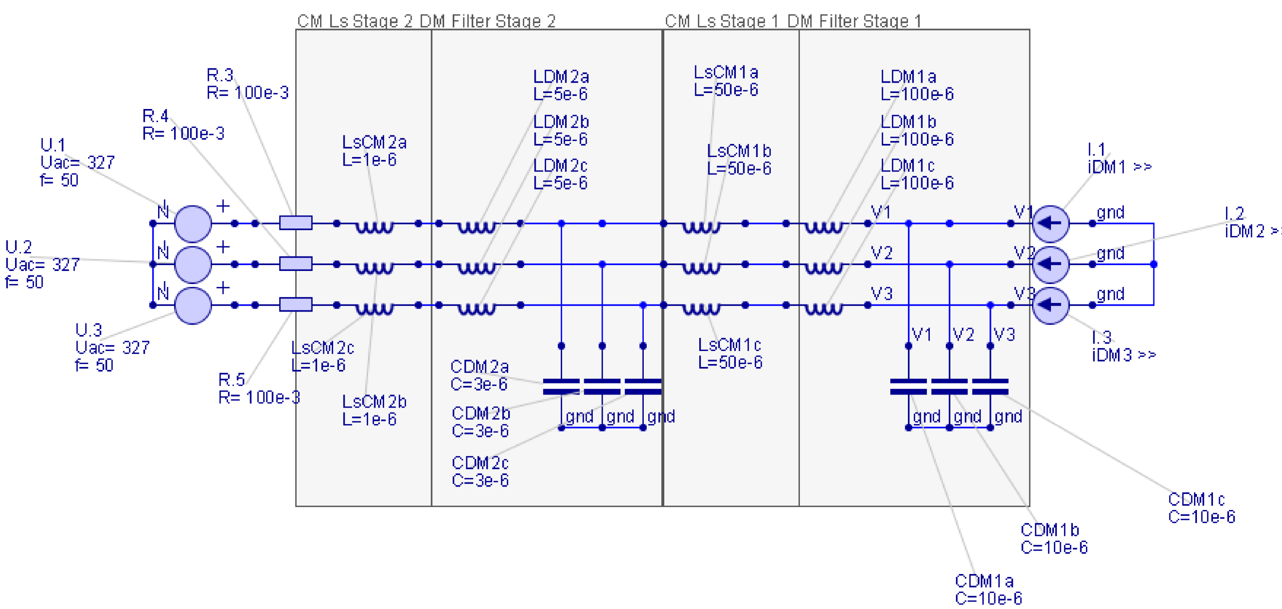


Figure 2: Two-stage differential mode filter and noise GeckoCIRCUITS simulation model used by the Three-Phase EMI Filter Optimizer.

Set up in this way, the DM filter model is more-or-less usable for any actual converter topology – it is **up to the user** to extract and provide a correct DM current waveform for the above filter model to use, as this waveform and its precise point of extraction from a converter simulation model may depend on the converter topology.

2.3 Common-Mode Simulation Model

The two-stage CM noise source and simulation model corresponding to the previously discussed DM model is given in Figure 3. Unlike the DM model, the CM model is bit more specific and dependant on the actual converter topology behind it. The supplied default model shown in Figure 3 is based on the three-phase buck-type SWISS rectifier[3], but should also be valid for many other three-phase buck-type topologies. Note the following necessary components:

- the common-mode filtering components for each stage: *L* and *C*
- the equivalent inductance of the DM inductors for each stage
- the **parasitic capacitances** C_{eq} and C_g
- common-mode **noise source**: a voltage source which represents the common-mode voltage of the converter. Like with the DM model, the common-mode voltage should be extracted from a simulation of the full converter, saved to a file, and then loaded to this voltage source.

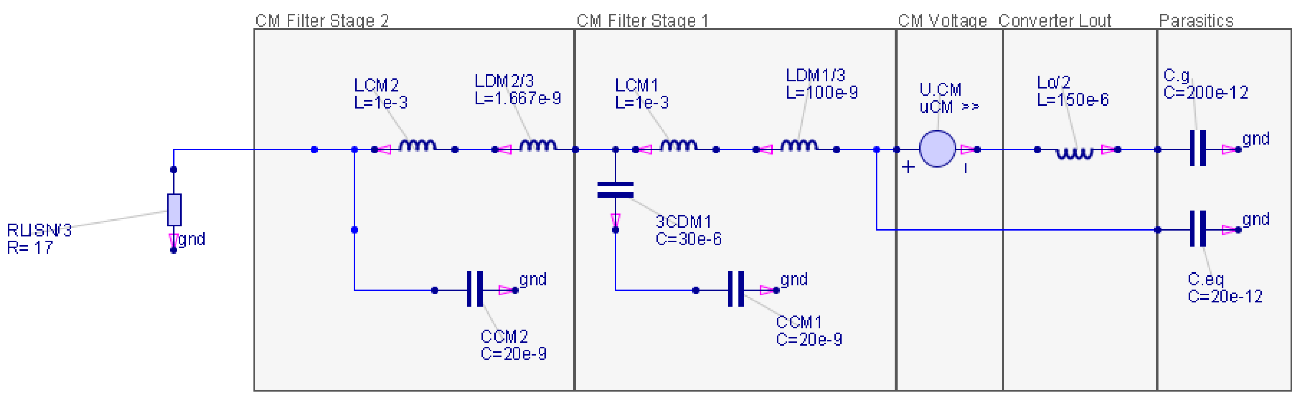


Figure 3: Two-stage common-mode filter and noise GeckoCIRCUITS simulation model used by the Three-Phase EMI Filter Optimizer. Shown is both a stage where the common-mode capacitors are in series with the differential-mode capacitors, and where they are not.

The following are optional components, which may be required based on your converter and/or filter topology:

- the scaled converter output inductance kL_{out}: this is necessary for buck-type topologies, but should be omitted for e.g. boost-type topologies
- the equivalent capacitance of the differential-mode capacitors for each stage: this is required if you select a filter topology where the CM capacitors are in series with the DM capacitors (this is configurable on a by-stage basis). If you select a configuration where they are not in series, it does not mean that you must remove this element from the simulation model – the Optimizer will automatically omit it when it performs the simulation.

More information on how these components influence the CM filter attenuation calculation will be given in subsequent sections. Note that in the default model, C_{eq} represents the parasitic capacitances from the semiconductors to heat sink, and the power connection terminals to earth, while C_g represents the parasitic capacitances from the positive and negative bus-bars and the load to earth. For the converter in your application, these elements may represent some other quantities (for example, they may be exactly "flipped" from the default assumption). It is the **user's responsibility** make sure that the CM filter and noise model is correct, and also to provide a correct common-mode voltage noise waveform.

3 See http://www.pes.ee.ethz.ch/uploads/tx_ethpublications/04_SWISS_Rectifier_APEC2012.pdf
T. Soeiro, T. Friedli, J. W. Kolar, "SWISS Rectifier – A Novel Three-Phase Buck-Type PFC Topology for Electric Vehicle Battery Charging", Proceedings of the 27th Applied Power Electronics Conference and Exposition (APEC 2012), Orlando Florida, USA, February 5-9, 2012.

2.3 Model Set-Up Tab

When the optimizer is started, the "Model Set-Up" tab is shown first. This is shown in Figure 4. Just below the schematic, is the "Number of filter stages to model" spinner selector. For each different stage configuration, there must exist a set of two simulations models: a CM one and a DM one. The number selected in this spinner (the default is 4) tells the Optimizer how many different models to expect to use, and what is therefore the absolute maximum number of stages that will be considered in an optimization. The maximum allowed number of filter stages is **15**. The 8 (4 DM, 4 CM, from 1-stage to 4-stage) simulation supplied with the Optimizer are loaded by default. By clicking the buttons "Add DM

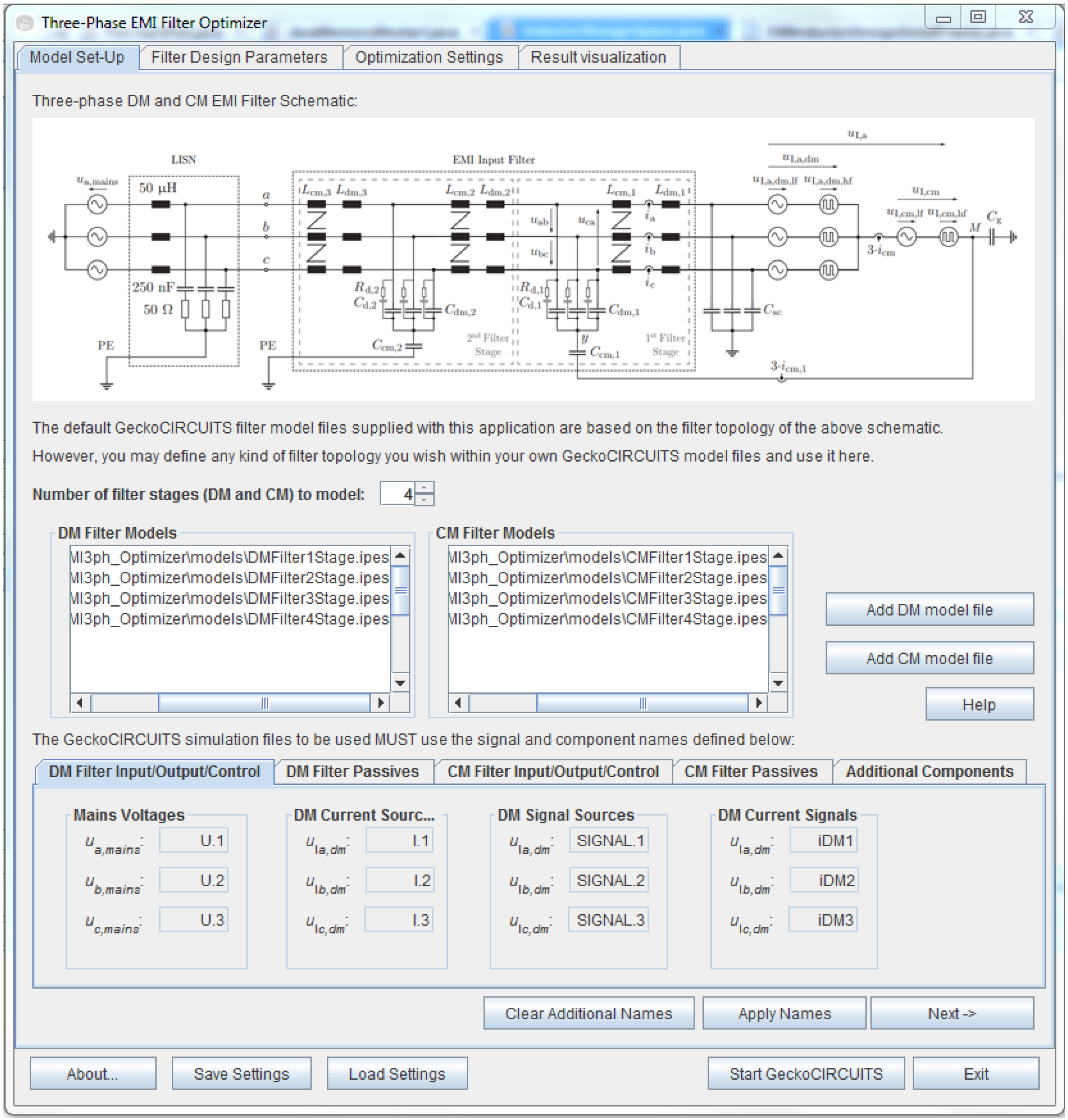


Figure 4: The Model Set-Up tab, starting screen of the Optimizer.

model file” and “Add CM model file” you can change the model files used, or add new ones for filters with more than 4 stages. For further information click the “Help” button to the right of the text boxes with model file names.

The simulations models must be set up correctly in order for the Optimizer to extract the necessary waveforms from them. The small tabbed pane below the model definition text boxes contains the required names of the simulation components. It contains the name that each corresponding component **MUST** have in the circuit simulation model in order for the Optimizer to correctly identify them. If you open the supplied default models in GeckoCIRCUITS, you will see that the names of the various componets are as given in this part of the Optimizer interface. If you modify the models for your purposes, or make ones to use, you must preserve this naming convention, otherwise the Optimizer will not work.

2.4 Additional Components' Names

The last tab of the small signal and component names tabbed pane in the lower part of the screen allows you to define additional components of the filter circuits. This is shown in Figure 5. For example, EMI filters often include damping elements, which are not included in the default models. The additional components are divided into four classes: DM filter single elements, DM filter per stage components, CM

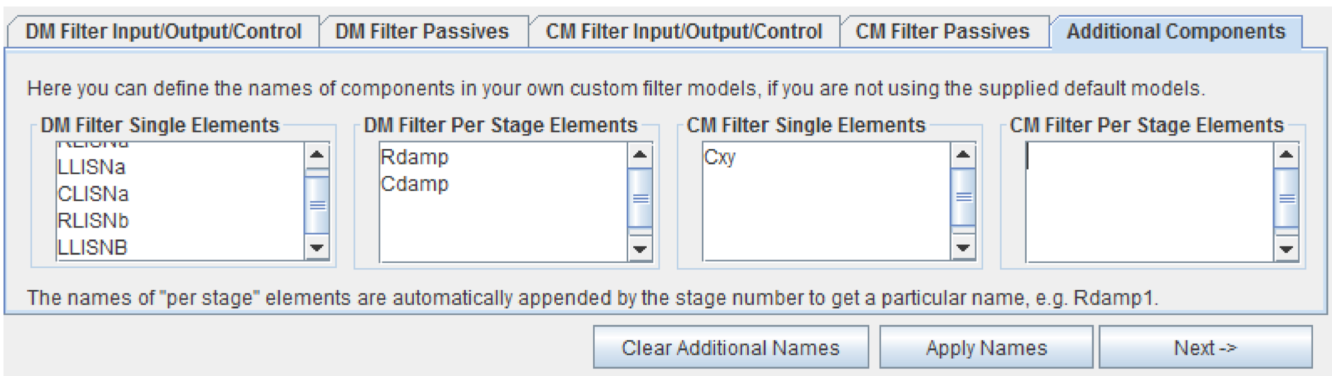


Figure 5: Tab for defining the names of additional components in the filter simulation models.

filter single components, and CM filter per stage components. Single components are those which appear only once in the simulation models. For example, if you wish to add LISN components to the DM filter model, or extra parasitic capacitances to the CM filter model. These components should have the exact same name in the simulation models as defined by the user in this tab. The “per-stage” components are those which appear in each stage of the filter. For example, damping resistors and capacitors. The names given by the user are therefore automatically appended by the stage number to produce the names which should be present in the simulation models. For example, if the user defines a DM per-stage additional component called “Rdamp”, the e.g. three-stage DM filter model must contain components Rdamp1, Rdamp2, and Rdamp3.

2.5 A Note on CM Feedback Capacitors

Many common-mode filter designs include a feedback capacitor between the filter and a point at the converter's output. This is for example shown in the filter schematic which is at the beginning of this manual, and also on the opening screen of the Optimizer. In some cases, the feedback capacitor is actually the capacitor of the first CM stage, as shown in this schematic. In other cases, it is a separate component which is not considered part of the first CM filter stage “proper”. A feedback capacitor may be beneficial for reducing CM noise (and therefore the required CM attenuation of the filter), or it may have an opposite effect and be detrimental, all depending on the size of the parasitic capacitances in the system.[4] However, determining analytically which is the case for a particular application and set of

4 See previously mentioned paper by D. O. Boillat et al.: http://www.pes.ee.ethz.ch/uploads/tx_ethpublications/24_Volume_Minimization_ECCE_USA_Boillat.pdf

parasitic components is not straightforward: when the CM capacitor of the first filter stage is connected as a feedback capacitor, the transfer function of the first filter stage becomes very complicated; a universal simplification is not possible because which factors of it are dominant depends on the relative sizes of the parasitics, the feedback capacitor, and the first stage common-mode choke. Solving the full transfer function (i.e. extracting required values of *L* and *C* for a given attenuation) is possible only numerically. For this reason, the Optimizer program does **NOT** allow the connection of the capacitor of the first CM filter stage (or any CM filter stage) in a feedback manner. The CM capacitors of each filter stage **must** be connected to earth. If you wish to include a "stand-alone" CM feedback capacitor in your models, set it to a constant value, and account for its effect in the required attenuation/noise to be attenuated given as input to the Optimizer (this is covered in the next section).

3. Global Parameters

The next tab of the Optimizer window is "Filter Design Parameters". This tab is divided into several sub-tabs. The first one "Global Parameters". Here is where the attenuation/noise, design frequency, input voltage (mains), and ambient temperature are defined. It is shown in Figure 6. The first option to select is the number of filter stages considered for a particular optimization run. This can be defined as a range, or fixed to a constant value. Next to this set of selection spinners is a text box to specify the ambient temperature in degrees Celsius. This is the ambient temperature in which the filter is assumed to operate, and is used when calculating inductor losses. Next, the design frequency of the filter must be specified. This is the frequency at which the filter attenuations are calculated. The user can specify this frequency directly, or, can instead enter the switching frequency of the converter the filter is being designed for, and the Optimizer will automatically determine the design frequency.

3.1 Specifying the Required Attenuation

The user can enter the required DM and CM attenuation directly, in dB, if known. Alternatively, the DM and CM noise can be entered in dBμV along with a set of emissions limits and a safety margin. From this, the Optimizer will automatically calculate the required attenuation for the DM and CM filters. Given in a menu is a set of common emissions limits that can be selected, but a custom set of limits is also defined. As the given limit values are valid up to 1 Mhz, the "custom" option should be used if the spectrum beyond 1 MHz is being considered. The attenuation of the DM filter is calculated using the simplified equation:

$$Att_{DM}(dB)=\sum_{i=1}^{n} 20\log(4\pi^2 f_D^2 L_{DMi} C_{DMi})$$

where *n* is the total number of stages and *i* a particular stage. The attenuation of the CM filter is calculated using the simplified equation:

$$Att_{CM}(dB)=Att_{CM1}(dB)+\sum_{i=2}^{n} 20\log(4\pi^2 f_D^2 L_{CMi} C_{CMi})$$

where the attenuation of the first filter stage is, for the case where a scaled converter output inductance is present in the model (e.g. buck-type converter topologies), is calculated by

$$Att_{CM1}(dB)=20\log\left(\frac{4\pi^2 f_D^2 kL_{out} C_g C_{eq}+\dfrac{2\pi f_D kL_{out} C_g}{1/3 R_{LISN}}+C_g+C_{eq}}{C_g}\cdot 4\pi^2 f_D^2 L_{CM1} C_{CM1}\right)$$

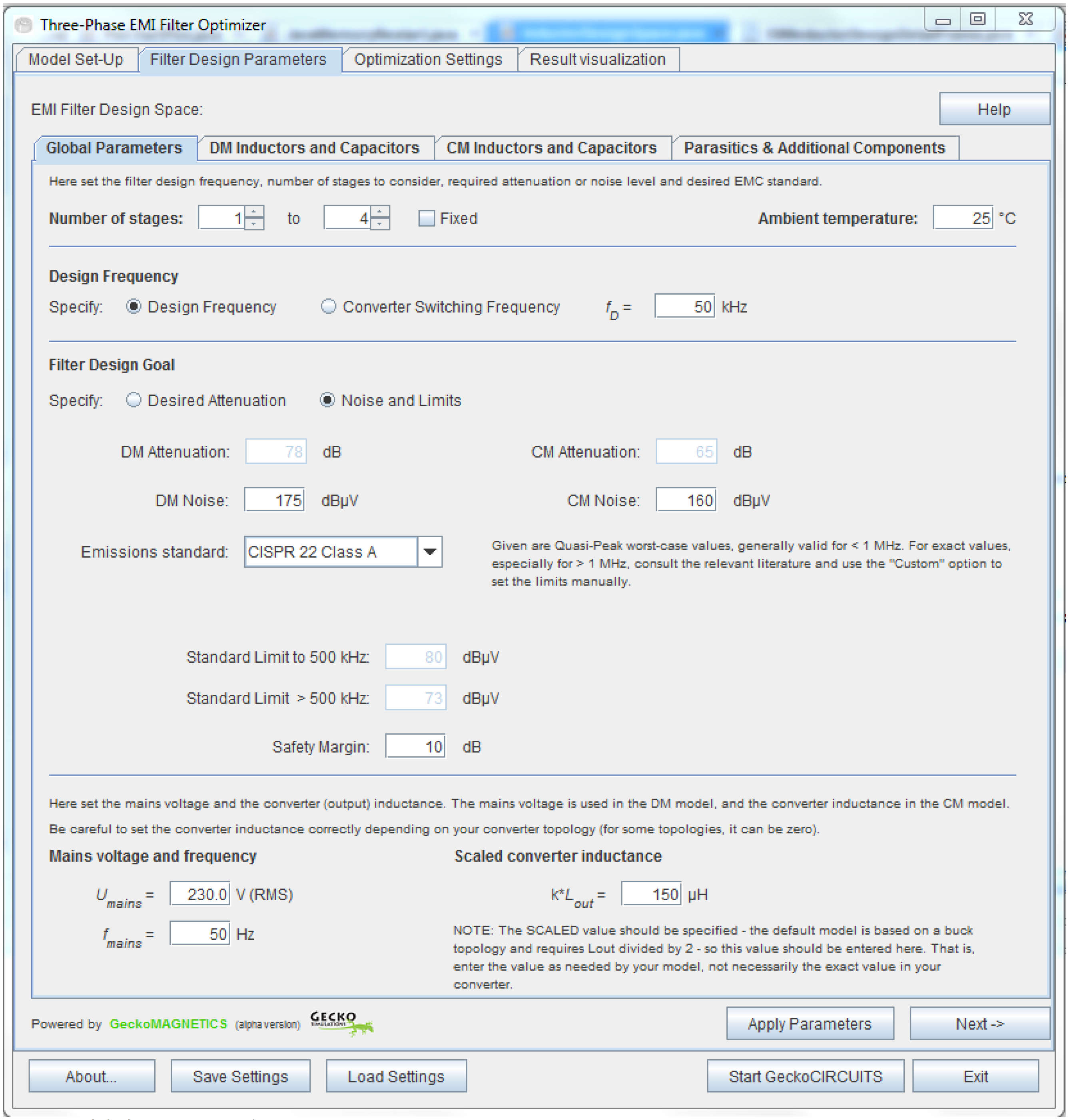


Figure 6: Global Parameters Tab

and when the scaled converter output inductance is not present in the model (e.g. boost-type converter topologies), is calculated by

$$Att_{CM1}(dB)=20\log\left(\frac{C_g+C_{eq}}{C_g}\cdot 4\pi^2 f_D^2 L_{CM1} C_{CM1}\right)$$

The above equations show the importance of the parasitic elements in calculating the CM filter attenuation. The user should do therefore do his best to calculate, estimate, or measure these values. However, if this cannot be done, and the CM noise or required attenuation given as input to the

Optimizer is based simply on a measurement of the entire converter, the parasitics values should be entered so as to cancel out and not affect the attenuation calculation. For example, for the case where not converter output inductance is considered, if C_{eq} is selected to be much smaller than C_g, in that case $(C_g + C_{eq}) / C_g \sim= 1$, and the attenuation is then effectively calculated on the *LC* values only, as with the DM filter. Therefore, please be very careful when specifying the required CM attenuation, and make sure it corresponds to the C_g, C_{eq}, and kL_{out} values that you enter.

3.2 Input Voltage

At the bottom of the Global Parameters tab, besides the aforementioned scaled converter inductance, the RMS mains voltage and frequency must also be entered. The default is a standard European 230 V, 50 Hz network. Please make sure to specify this correctly for your application. The values entered here will be set in the GeckoCIRCUITS simulation model.

4. DM Inductors and Capacitors

The next sub-tab of the Filter Design Parameters tab is "DM Inductors and Capacitors". Here the ranges of the values for the differential-mode filter filtering inductors and capacitors is defined, as well as the design spaces for the actual implementation of these components. The use of design spaces is covered in Sections 6 and 7 of this manual. In this section the other aspects of this tab are covered. A screenshot of the DM Inductors and Capacitors tab is shown in Figure 7.

4.1 Maximum total differential-mode capacitacance per phase

First, the maximum total per-phase DM capacitance must be specified at the top of the screen. Set this value carefully on your maximum mains current phase displacement requirement, and/or other design constraints of your power converter for which you are designing the EMI filter. This is a hard constraint, but allow for a 20% tolerance due to a possible restriction in available component values (e.g. if you enter 13 μF, a closest possible actual implementation of a capacitor might be 14.1 μF), i.e. set this value to be somewhat lower than your absolute allowed maximum. **Generally, be always aware that the Optimizer is basically "blind" to the actual converter topology, and that you must carefully set constraints such as these in order to follow the general constraints of your overall design problem.**

4.2 DM Filter Stage 1

The first DM filter stage is often considered and designed separately from the remaining ones, for various reasons. In boost-type topologies, the inductor of stage 1 is actually the boost inductor, and therefore and integral part of the converter which needs to be designed according to the overall converter specifications. Also, regardless of the topology, the capacitors of stage can be treated as input capacitors of the converter, and need to have a certain value. In other cases, for e.g. control stability reasons, it is desirable for the first stage to comprise a certain percentage of the overall DM attenuation. For this reason, the parameters of the first stage are specified separately from the rest. Here the range of values to be considered for the first stage inductor can be specified (or this inductance can be fixed to a constant). The capacitance of the first stage can be set to be fixed to a given fixed value, or allowed to be freely selected by the Optimizer. Finally, the user can set the first stage to have a certain fixed proportion of the overall DM attenuation, assuring that e.g. 0.7 of the total required DM attenuation is supplied by the first stage.

The ability to set the first-stage DM capacitance to a constant is also provided in order to allow the user to set a minimum total DM capacitance – without this, the minimum is limited only by the smallest available capacitor in the design space greater than or equal to 1 nF.

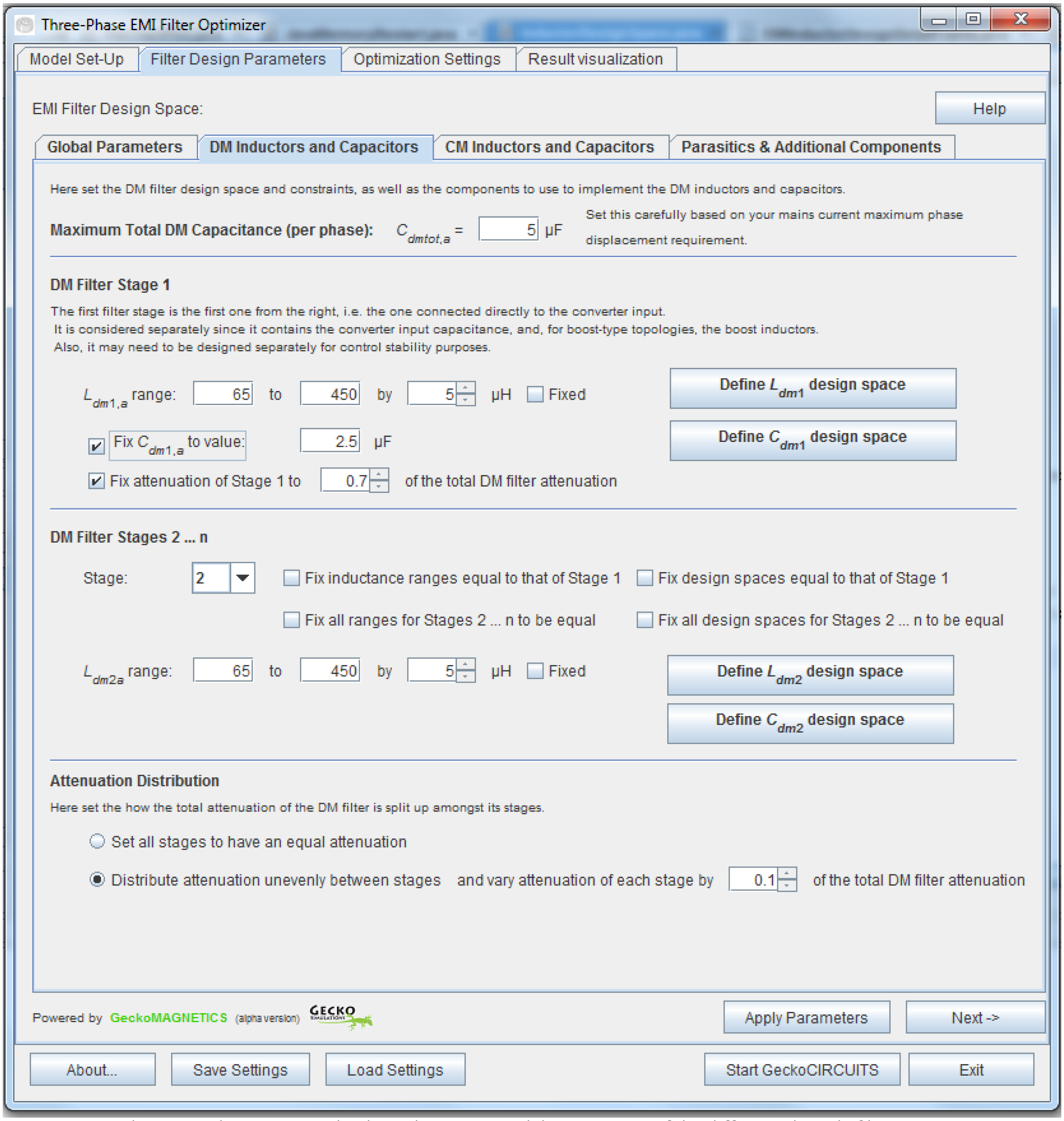


Figure 7: DM Inductors and Capacitors tab where the ranges and design spaces of the differential-mode filtering components are defined.

4.3 DM Filter Stages 2 .. n

Below the space where the first filter stage is specified, the remaining filter stages (if any) can also be specified. For each subsequent stage, the range of inductor values as well as the inductor and capacitor design spaces must be given. Be aware that the range given for the values is more of a guide or "suggestion" for the Optimizer, and will not necessarily be strictly followed (unless fixed to a constant value). What will be strictly followed is the required attenuation of a given stage and the maximum allowed capacitance per stage. In all cases where the inductance for a particular stage is not fixed, LC combinations with 100%, 50%, and 25% of the maximum allowed capacitance for that stage will always be considered, regardless of the specified range. Also, if a set constant inductance value is impossible (for example, it requires, for the required attenuation of that stage, a capacitance above the allowed maximum), a new inductance value will be calculated based on the maximum allowed capacitance for that stage.

A given stage can be selected from the drop-down menu.

Also, here you can set the ranges and/or design spaces for stages 2 .. n to be all the same (but different from those defined for stage 1), or identical to those already defined for stage 1.

4.4 DM Attenuation Distribution

At the bottom of the screen, you can select whether the attenuation is split equally amongst the stages (other than the possibly fixed attenuation of stage 1), or varied. If the attenuation of each stage is to be varied, the step (as a proportion of the total attenuation) by which to vary the attenuation must also be specified. When attenuations are varied from stage to stage, to reduce the size of the optimization problem , only combinations with decreasing attenuation per-stage are considered. For example, for a two-stage filter, where attenuation is varied by 0.1 of the total between the stages, the considered attenuation combinations will be

Stage 1	Stage 2
0.5	0.5
0.6	0.4
0.7	0.3
0.8	0.2
0.9	0.1

where 1.0 is the total DM attenuation. In the varied case, two possibilities for setting a maximum allowed capacitance for each stage are considered – one are combinations where the maximum allowed capacitance is distributed equally among the stages. For example, if the maximum allowed capacitance is 10 μF, in a two-stage filter the maximum capacitance of each stage will be 5 μF, regardless of the attenuation distribution. The second possibility are combinations where the maximum allowed capacitance per stage is “cascaded” according to the attenuation distribution. For example, for the maximum total DM capacitance of 10 μF, for a two-stage filter where stage 1 delivers 0.6 of the total attenuation and stage 2 0.4, the maximum allowed capacitance for stage 1 is 6 μF, and for stage 2, 4 μF.

5. CM Inductors and Capacitors

The next sub-tab is of the Filter Design Parameters tab is “CM Inductors and Capacitors”. It is very similar to the previous, DM tab. Here the ranges of the values for the common-mode filter filtering inductors and capacitors is defined, as well as the design spaces for the actual implementation of these components. The use of design spaces is covered in Sections 6 and 7 of this manual. In this section the other aspects of this tab are covered. A screenshot of the DM Inductors and Capacitors tab is shown in Figure 8.

5.1 Maximum Total CM Capacitance

The maximum total common-mode capacitance can be defined in two ways. It can either be specified directly, in nF, or instead the maximum leakage current to earth can be given. If the latter option is chosen, the Optimizer will automatically calculate the maximum total capacitance. When calculated automatically, a 20% tolerance is automatically assumed. Be likewise mindful when specifying the capacitance directly. This is a hard constraint which the Optmizer will always fulfill when searching for valid filter designs.

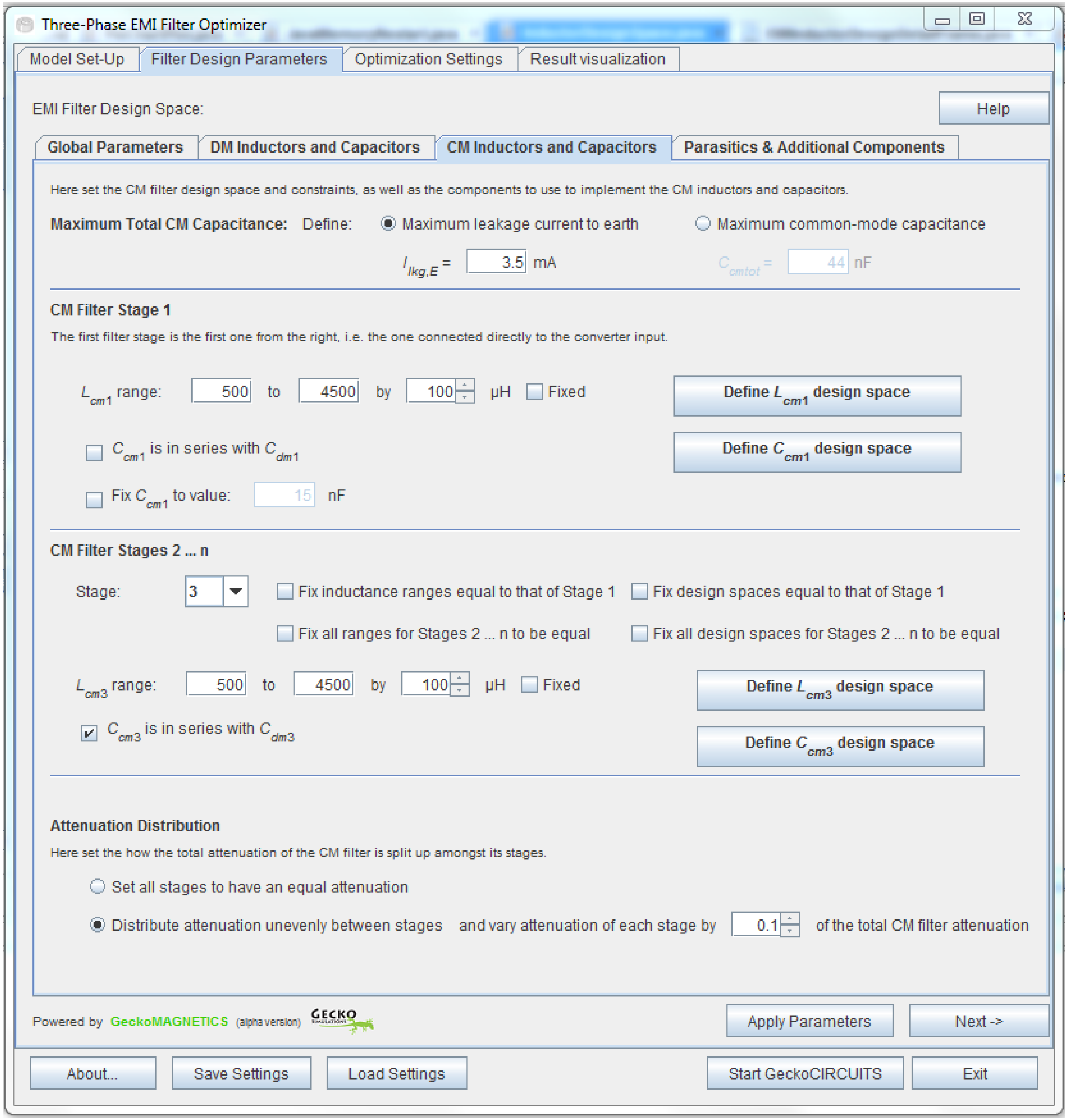


Figure 8: CM Inductors and Capacitors tab where the ranges and design spaces of the common-mode filtering components are defined.

5.2 CM Filter Stage 1

Like with the DM filter, the first stage of the CM filter is considered separately from the rest. Here the main reason is to allow the capacitance of the first stage of the filter to be set to a constant, thereby defining a minimum total CM capacitance. If the capacitance of the first stage is not fixed to a constant value, the minimum capacitance depends on the smallest available capacitor in the design space that is greater than or equal to 1 nF. A checkbox defines whether the capacitance of the first CM stage is in series with that of the first DM stage, or not.

5.3 CM Filter Stages 2 .. n

The remaning stages, if any, can be selected from the drop-down menu. For each stage, the range of common-mode inductance values can be specified, as well as whether the common-mode capacitance of that stage is in series with that stage's differential mode capacitance. Be aware that the range given for the values is more of a guide or "suggestion" for the Optimizer, and will not necessarily be strictly followed (unless fixed to a constant value). What will be strictly followed is the required attenuation of a given stage and the maximum allowed capacitance per stage. In all cases where the inductance for a particular stage is not fixed, LC combinations with 100%, 50%, and 25% of the maximum allowed capacitance for that stage will always be considered, regardless of the specified range. Also, if a set constant inductance value is impossible (for example, it requires, for the required attenuation of that stage, a capacitance above the allowed maximum), a new inductance value will be calculated based on the maximum allowed capacitance for that stage.

Also, here you can set the ranges and/or design spaces for stages 2 .. n to be all the same (but different from those defined for stage 1), or identical to those already defined for stage 1.

5.4 CM Attenuation Distribution

Like with the CM filter, here it can also be chosen whether the CM attenuation is distributed equally amongst the stages or not. Everything listed in Section 4.4 as applying for the DM fitler applies here as well for the CM filter.

6. Defining a Capacitor Design Space

For both the DM and CM portions of the filter, for each stage a capacitor design space must be defined. A **design space** defines the available components for the implementaion of a capacitor. When a "Define Cxxx Design Space" button is clicked, the capacitor design dialog shown in Figure 9 appears. There are three ways of specifying a capacitor design space: selecting (an) individual capacitor(s), selecting (a) capacitor series, or specifying a custom capacitor.

Note: Capacitors typically used in EMI filters have negligible leakage current. However, in the Optimizer the user has the option to define leakage current characteristics for capacitors, in order to provide a more full modelling experience if so desired, or if capacitors not typically used in EMI filters are necessary for a certain application.

6.1 Individual Capacitors

If the "Select individual capacitor(s)" radio button is selected, a capacitor can be selected from the drop-down list. Since a capacitor design space will typically be used to provide several capacitors of different capacitance values, by clicking "Multiple Selection" the user can select several individual capacitors to be in the design space. Selecting all of the capacitors in the Optimizer's database is possible. Clicking on the "Details" button opens the GeckoDB database, a custom beta version of which is used by the Optimizer. The Capacitor tab of GeckoDB is then displayed, as shown in Figure 10. This shows all of the capacitors which come preloaded with the Optimizer. It is possible to modify the capacitors in the database, delete (be careful! There is no way to recover deleted data!) them, or add new capacitors which will then be accessible to the Optimizer. Each entry is meant to be a physical, actual capacitor component – hence the total volume, voltage rating, capacitnce, and dissipation (ESR) factor has to be entered. Optionally, a leakage current characteristic can also be entered. Capacitor datasheets are also stored in the database and can be viewed by clicking "Datasheet". The voltage rating is entered as given in the datasheet, which means that for EMI suppression capacitors it is AC, i.e. volts RMS.

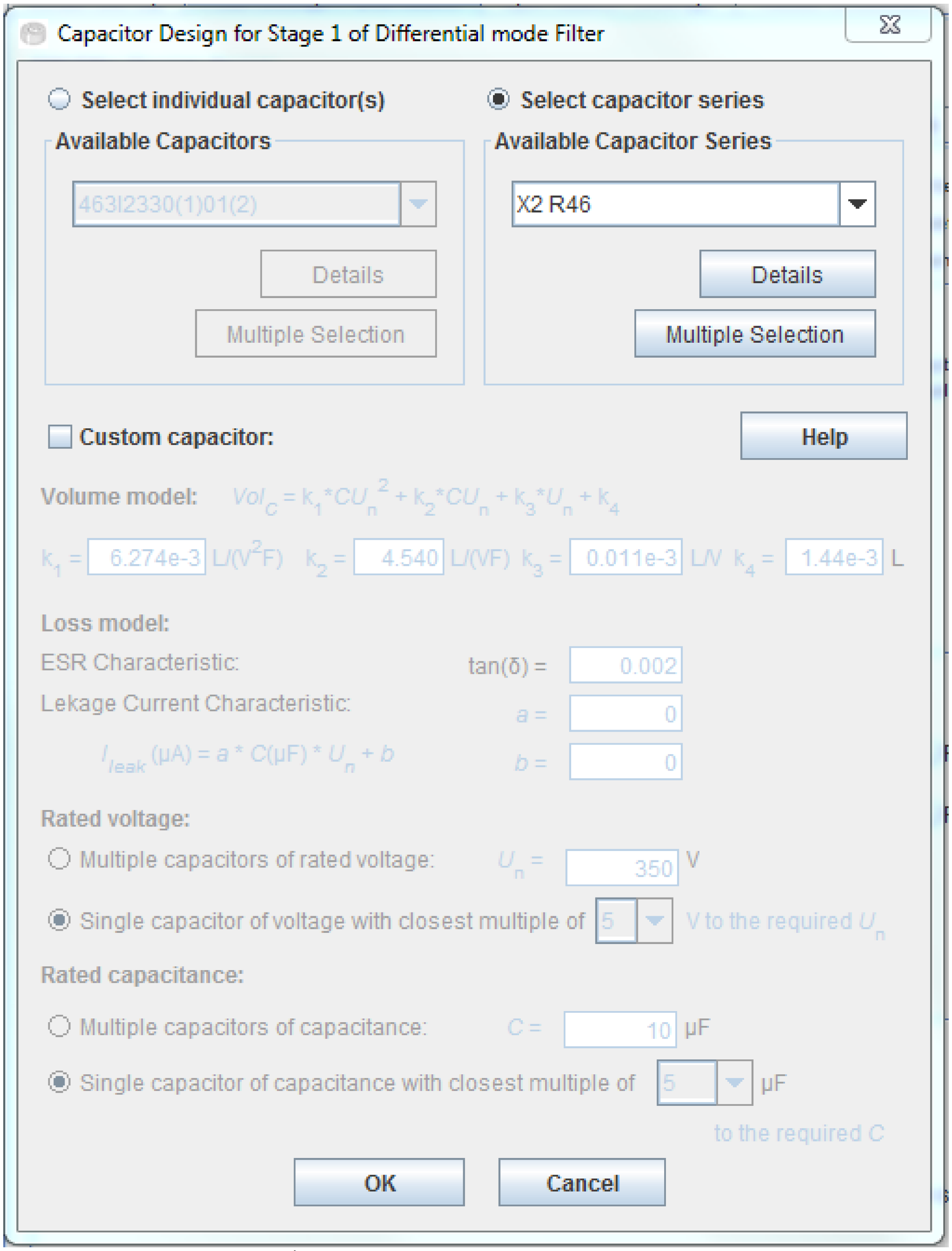


Figure 9: Capacitor Design Dialog

6.2 Capacitor Series

Typically, capacitors are produced and grouped into series, based on their application (e.g. X2 EMI suppression capacitors), voltage rating, and range of capacitance values. An engineer might have one or two appropriate series at his disposal, and must choose the best capacitor from amongst those for his or her design. For this reason, all of the capacitors supplied in the Optimizer's database are grouped into series, and this provides a more practical way of defining a capacitor design space. As seen in Figure 9, this is the default approach for the capacitor design space specification. If the "Select capacitor series" radio button is selected, a single capacitor series can be selected from the drop-down menu, or multiple capacitor series can be selected for the design space by clicking on "Multiple Selection". This has the same effect as selecting all of the capacitors in a series individually, but is more practical since a series

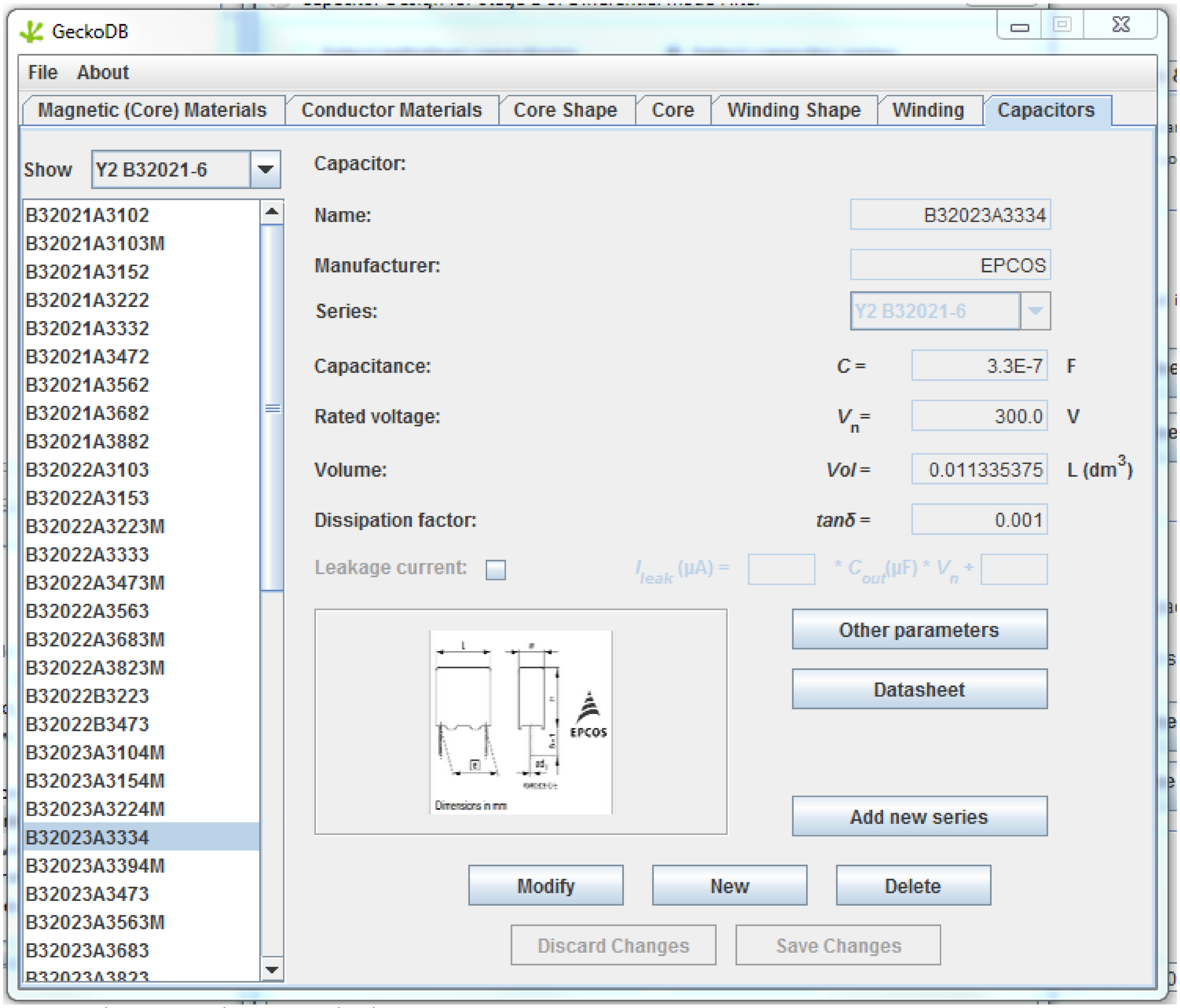


Figure 10: The Optimizer's capacitor database.

may contain hundreds of capacitors. As can be seen in Figure 10, the capacitors can be displayed by series in GeckoDB (by using the drop-down box in the upper-left corner of the screen), and for each individual capacitor the series which it belongs to can be defined. If the "Details" button in the capacitor series section of the design space dialog is clicked, the Capacitor Series dialog of the GeckoDB database will be opened, as shown in Figure 11. This shows the pre-loaded capacitor series supplied with the Optimizer – each capacitor in the database is assigned to one. The only required piece of information for a capacitor series is its name. However, dissipation factor (ESR) and leakage current data, dependant on the rated voltage, can also be entered. This simplifies entering individual capacitors of that series into the database, as when a new capacitor of a given series is entered, a dissipation factor and/or leakage current coefficients will be automatically suggested by the database based on the series data for that capacitor's rated voltage.

Note that datasheets are entered for individual capacitors, not capacitor series. However, as manufacturers usually supply datasheets on a per-series basis, in GeckoDB you will not find a datasheet for each individual capacitor. Rather, when you select a particular series for display in the screen of Figure 10, typically the capacitor at the top of the list will contain the datasheet for the entire series.

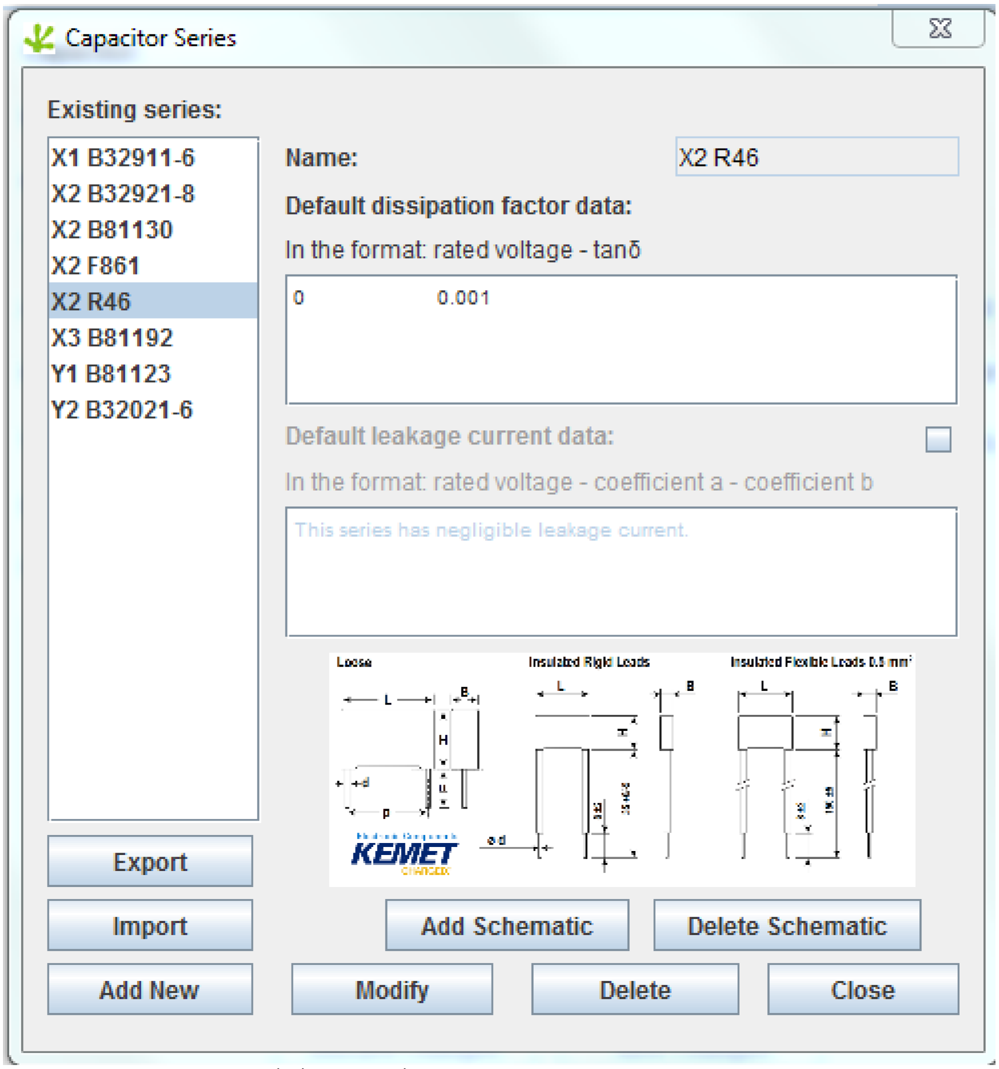


Figure 11: Capacitor series dialog in GeckoDB.

6.3 Custom Capacitors

The third option for defining a capacitor design space is to define a custom capacitor. This option is given in order to allow the user to quickly define a capacitor design space without having to enter tens or hundreds of capacitors into the database first. However, if you plan to use certain capacitors often in your designs, it is recommended that you take the time to enter them into the database.

The various things that should be defined are shown in the lower half of the capacitor design dialog of Figure 9. By selecting the “Custom capacitor” check-box these various fields become active. First, the volume model must be defined – this is a quadratic equation which defines the relationship between the capacitance and rated voltage of a capacitor and its total size. If you wish to user a simpler model, you can set some of the constants to zero. Next, the loss model must be defined through the input of a dissipation factor for ESR calculation and leakage current model coefficients. To ignore leakage current as is the case for most EMI filter capacitors, set these coefficients to be zero. When it comes to defining the rated voltage and the capacitance of the custom capacitor, two options are available. One is to define a single capacitor of a given rating and/or capacitance value. The necessary filter components will then be built (if possible) out of this one capacitor. The other option is to in essence define a series, by defining the multiple in which capacitors are available. For example, if the user selects for the rated voltage, a single capacitor rated for 350 V, while for the capacitance multiple capacitances available in multiples of

5 μF, the resulting design space will contain capacitors with values 5, 10, 15, 25, … μF, all of them with a rated voltage of 350 V.

6.4 How the Optimizer selects capacitors

Due to the negligible leakage current and generally small dissipation factors for EMI suppression capacitors, capacitor losses will be primarily dependent on the capacitance value and frequency of operation. Therefore, the Optimizer selects capacitors primarily on volume. The following selection criteria are used:

- For a given capacitance, the Optimizer attempts to find the closest implementation (within 20% at most) achievable with the components in the design space
- If possible, only combinations with less than 4 capacitors in series are considered, in order to simplify the problem and allow for a practical implementation
- Placing capacitors in series is NOT considered: therefore if a design space does not contain the required rated voltage, no valid capacitor design can result
- Out of the possible combinations of capacitor components for the given capacitance, the one with the lowest volume is selected
- If there are several combionations possible with the same lowest volume, the one with the lowest dissipation factor (and therefore losses) is selected.

7. Defining an Inductor Design Space

For both the DM and CM portions of the filter, for each stage an inductor design space must be defined. A **design space** defines the available components for the implementaion of an inductor. When a “Define Lxxx Design Space” button is clicked, the inductor design dialog shown in Figure 12 appears. There are many options to define an inductor design space. In addition, the user can define how the inductor is selected from the design space.

The Optimizer uses a custom-tailored alpha version of **GeckoMAGNETICS**[5] to model and simulate inductors.

7.1 Core Material

One magnetic core material can be defined for the design space. The list of available core materials is given in the first drop-down menu in the upper left corner of the inductor design dialog. Clicking on “Details” below this will open GeckoDB, and show the magnetic materials stored in the Optimizer's database. This is shown in Figure 13. Here the user can define additional materials to use. The various parameters of the magnetic materials must be entered. The first two basic ones are the initial relative permeability and the maximum permissible flux density. The second one is especially important, as it is used to reject unsuitable inductor designs. The inductance of an inductor using a particular core material will be calculated using the entered initial relative permeability, unless B-H curves of μ-H curves are entered for the material. This can be done by clicking the appropriate buttons on the screen of Figure 13, which will open the appropriate dialog for entering such curves, as shown in Figure 14. “Initial” curves are not loops, and are typically extracted from datasheets. “Extracted” curves are B-H loops resulting from actual measurement of the material properties. For each curve, a temperature must be entered. The Optimizer will select the closest curve based on the operating temperature calculated from simulation. Initial curves will be used first, and if not present, then extracted curves, for current-dependant inductance characteristic calculation of the filter inductors.

A very important aspect of the core material definition is its **loss data**. Each material in the database has a **Loss Map**, a set of operating points defined by flux density, frequency, temperature and DC bias where

5 The full commercial version of GeckoMAGNETICS is a product of Gecko-Simulations AG. See more at: http://www.gecko-simulations.com/geckomagnetics.html

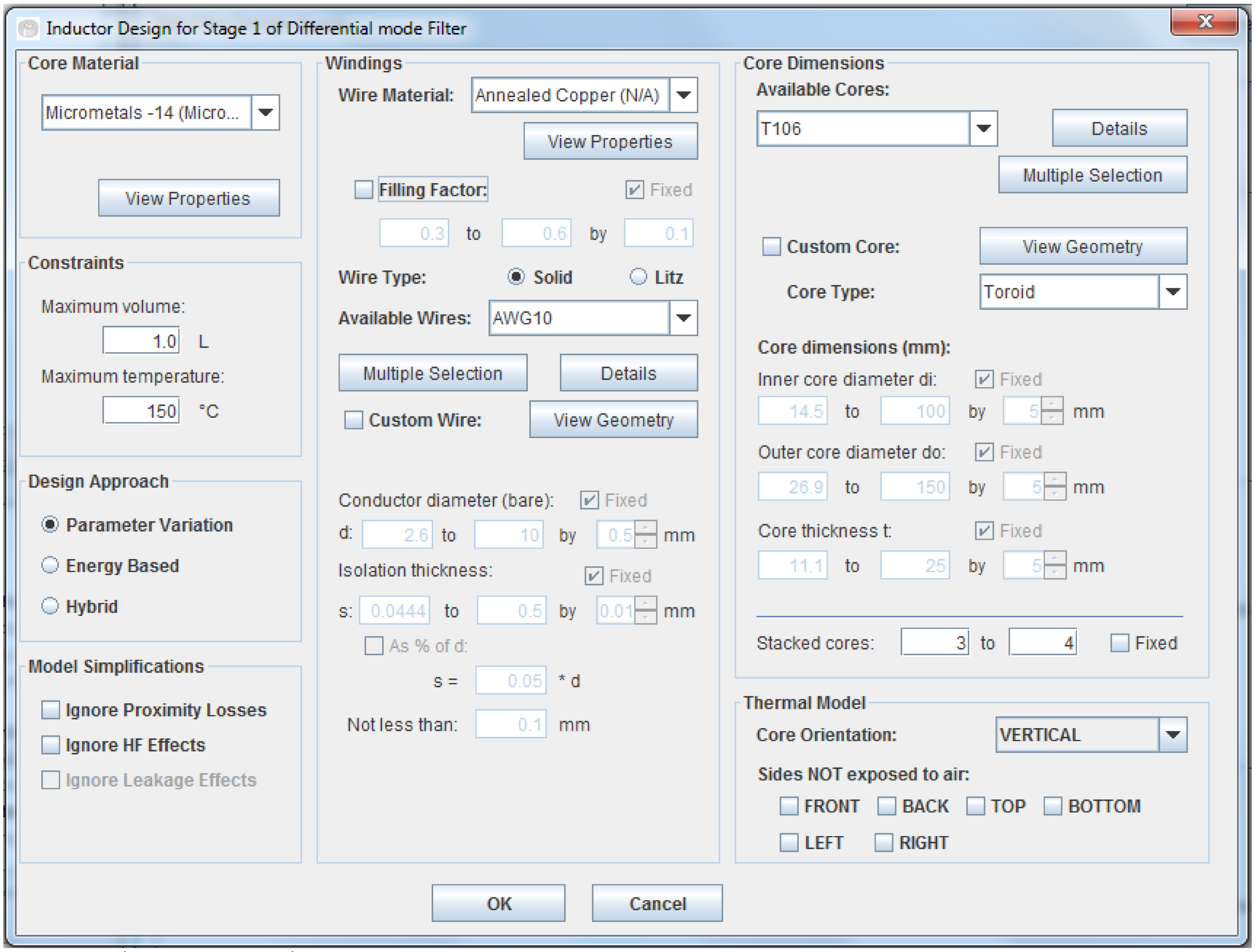


Figure 12: Inductor Design Dialog.

such operating point a volumetric loss (W/m³) is defined. This data is used to calculate the appropriate Steinmetz parameters for calculating inductor losses during simulation. For more information on the loss map concept, please see the relevant papers by J. Muehlethaler et al.[6] By clicking "View Loss Measurements", the Loss Map dialog is displayed, as shown in Figure 15. The loss map data may be from actual measurements, but may also be entered based on datasheet curves. If you are adding your own material to the database, it is highly recommended that you enter at least loss curves from the datasheet, if no measurements are available.

If the Loss Map is empty, Steinmetz parameters can be entered directly. In that case, this one set of Steinmetz parameters will be used at all times, regardless of inductor operating conditions.

7.2 Windings

For the wires to be used for the inductor windings, it is possible to define one conductor material. Conductor materials are also stored in the database, where they can be modified, or new ones added. Clicking "Details" below the Wire Material drop-down menu will open the database to the Conductor Materials tab.

It is possible to define wires in three ways: by pre-selecting predefined actual wires from the database, by

6 For example, J. Mühlethaler, J. W. Kolar, A. Ecklebe, "Loss Modeling of Inductive Components Employed in Power Electronic Systems", Proceedings of the 8th International Conference on Power Electronics (ECCE Asia 2011), The Shilla Jeju, Korea, May 30-June 3, 2011. See: http://www.pes.ee.ethz.ch/uploads/tx_ethpublications/07_Loss_Modelling_ECCEAsia2011_01.pdf

specifying a custom wire with a range of values for each wire dimension (e.g. strand diameter and number of strands for a litz wire), and to specify a fill factor range. Predefined round solid and litz wires

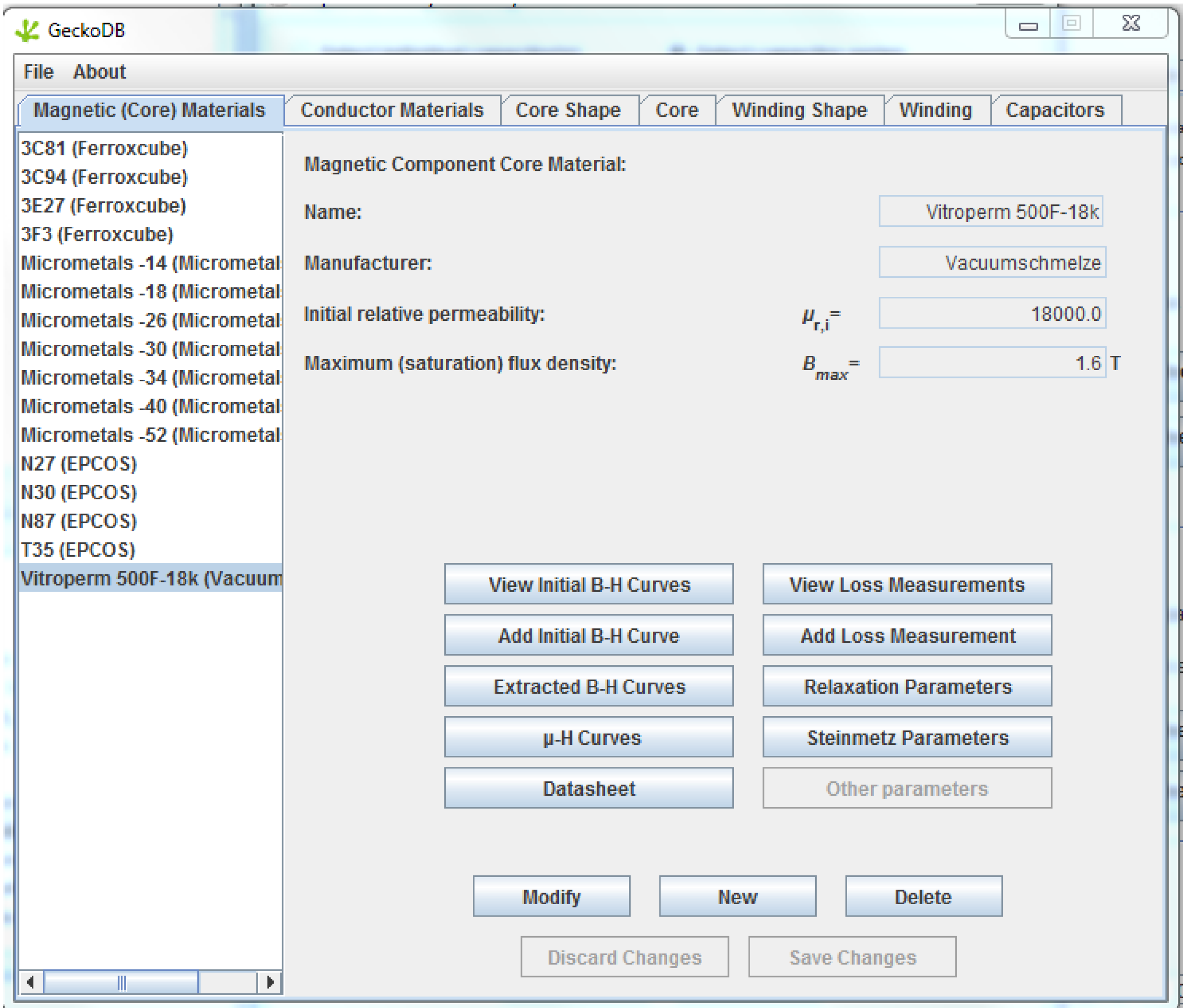


Figure 13: Core Materials in GeckoDB.

are stored in the database as well, and can be viewed by clicking the "Details" button underneath the wire selection drop-down menu. By clicking the button "View Geometry", the database will open to the Winding Shape tab, where the explanation of the various wire dimensions will be given, with a diagram. A single predefined wire can be selected, or several of them simultaneously by clicking on "Multiple Selection",

If specifying a fill factor instead of individual wires to use (whether from the database or custom), a custom wire will be created based on the fill factor and the window size of the utilized core(s). Even with the fill factor, some wire dimensions must be specified – for solid wires, the isolation layer thickness (which can be given as a range of values or a percentage of the total wire diameter), and for litz wires, the strand diameter and the compact factor (from which the number of strands is calculated, given the total diameter calculated from the fill factor, and the specified strand diameter). Note that not all wire types and definition types are available with all design approaches (see subsection on Design Approaches for details).

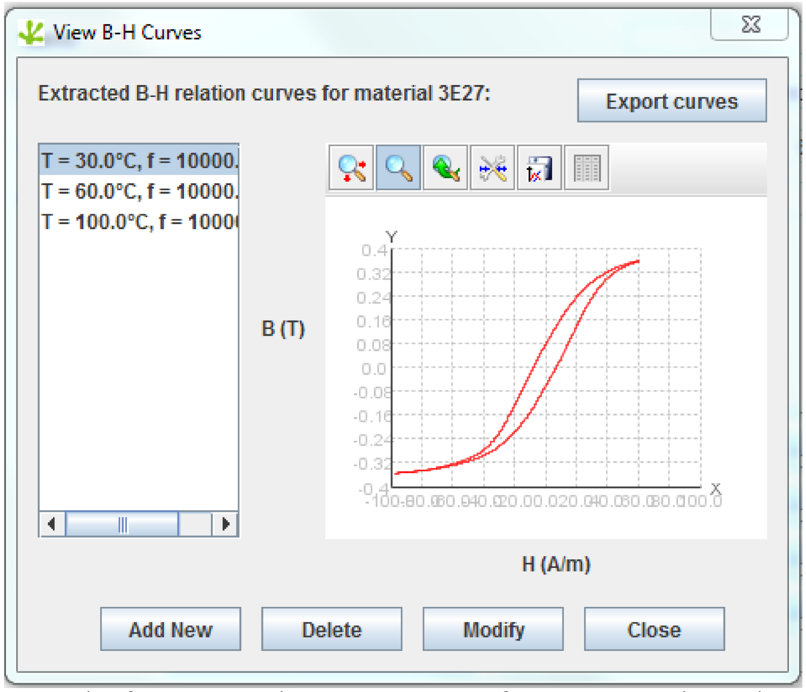


Figure 14: Dialog for viewing and entering B-H curves for a core material in GeckoDB.

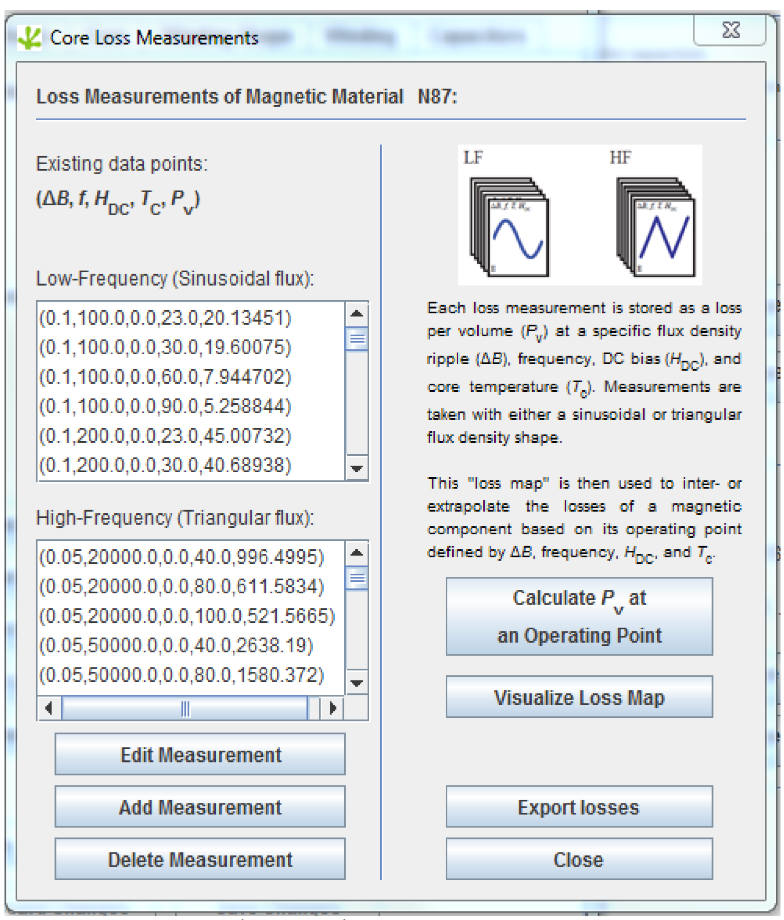


Figure 15: Loss Map Dialog in GeckoDB

It is not possible to mix wire types in a design space, i.e. only litz or only solid wires can be used.

7.3 Cores

For the cores to be added to the design space, the user can select predefined cores from the database, or define a custom set of core dimensions. The following core shapes are available:

- For DM inductors: Toroids and EE-Cores with 3 air gaps
- For CM inductors: Toroidal cores only.

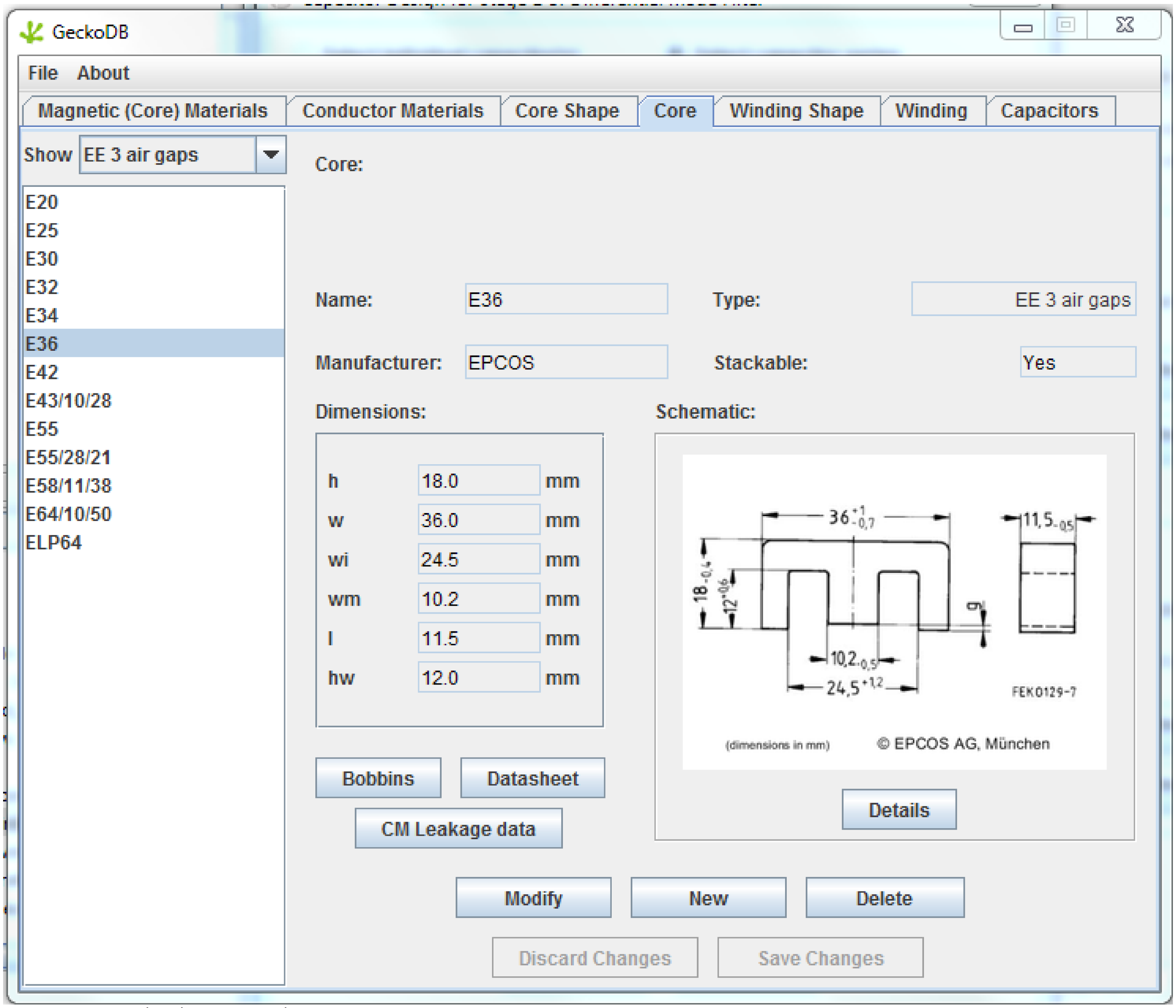


Figure 16: Core database in GeckoDB.

It is not possible to mix core types in a design space, i.e. only toroidal or only EE cores can be used in a single design space. Single predefined cores can be selected from the drop-down menu, or several can be selected by clicking “Multiple Selection”. If the “Details” button next to the core selection drop-down menu is clicked, the database will open and show the available cores, with all details, as seen in Figure 16. Here, you can enter additional cores beyond those supplied with the Optimizer. Any new core entered in the database will become automatically available in the inductor design dialog. Clicking the “View Geometry” button next to the “Custom Core” checkbox will open GeckoDB's Core Shape tab, where each of the dimensions of the specified core type is explained and shown on a diagram.

Be aware that for EE cores, depending on the design approach, you might need to specify a range of values for the air gap. To do this, scroll down to the bottom of the dimensions sub-panel when the EE cores shape is selected, and check if the air gap section is enabled for user input. If it is, specify a range of air gap values for the design space.

If specifying a custom core, the range of values for each core dimension must be entered.

For both custom and predefined cores, the number of cores stacked to produce the inductor must also be entered.

The custom core option is envisioned for finding an optimum core shape and size beyond those offered in standard core catalogues. If, on the other hand, you simply wish to use cores other than those supplied by the Optimizer by default, just add those cores to the database and use them as predefined cores.

7.4 Thermal Properties

The Optimizer considers only natural convection cooling for the EMI filter inductors, as this is usually how EMI filters are cooled. Within this thermal model, it is possible to set which sides of the inductor are exposed to air and thus conduct heat into the ambient. By default, all sides are exposed. If a side is checked in the inductor design dialog as being not exposed to air, the thermal model of the inductor will not consider any heat transfer through this side of the component. The sides are named from the core shape diagrams given in the database – the “front” side is the side whose cross-section is given in the diagram, and the “back” side is that which opposite from it, and not visible in the diagram. From this the definitions for the “top”, “bottom”, “left”, and “right” side of the inductor naturally follow.

The core orientation relative to the gravity direction also needs to be specified, as this affects the thermal model parameters. A “vertical” orientation is one in which the “bottom” side is facing down (i.e. “on the ground”), whereas a “horizontal” orientation is one in which the “front” side is facing down.

7.5 Constraints

For each design space, the maximum allowed temperature and volume must be specified. Inductors resulting from this design space that exceed this volume and/or temperature will be rejected by the Optimizer.

7.6 Design Approach

An important setting is the design approach option. Three different approaches to inductor design are possible. The first is **Parameter Variation**. As its name says, it a straightforward simple approach where the Optimizer tries out all the different values set in this dialog – i.e. the given range of core parameters or predefined cores, the different range of fill factors and/or wire parameters, or predefined wires. In short, the parameters of the design space are varied within the defined ranges until an optimum inductor design is found.

The second is the **Energy-Based** approach. This mostly follows an approach for core and wire selection used by engineers in traditional, non-computer-aided design, called the *Core Geometry Approach*, as defined in the *Transformer and Inductor Design Handbook*[7]. Based on the core's core geometry coefficient and area product, and the requried inductance and envisioned operating point, a core is selected to build the inductor from. This approach reduces the amount of data the user must enter – for example, for custom cores, only some of the inductor dimensions can be entered, as others are calculated directly by this approach. The number of stacked cores (in this case, the range entered by the user serves a min./max. constraint, not a range to be iterated over) and the air gap for EE cores for example, are calculated directly by this approach. This approach should increase the speed with which core are selected for example, since unsuitable cores should be eliminated at the beginning, without having to

7 Col. Wm. T. McLyman, *Transformer and Inductor Design Handbook*, CRC Press, 2011.

simulate them (as would be the case with the Parameter Variation Approach). However, the Energy-Based approach may not hit the optimimum in every design space, since it does not explore every corner of it, but instead directs the search based on sometimes simplifying assumptions.

For this reason, a third approach is available, the **Hybrid** approach, which combines the Energy-based approach with Parameter Variation. With the Hybrid approach, an initial set of parameter values is calculated using the Energy-based approach's methodology, but then these values are varied, in order to explore the design space in a range around the component designed by the Energy-based approach. Therefore, whereas the Energy-based approach returns one inductor for a given set of design parameters and constraints, the Hybrid approach returns a set from which the best is selected. The Hybrid approach is therefore more time-consuming, but more likely to yield results closer to the exact optimum in a design space.

Note that the use of litz wires is available only when using Parameter Variation, and that with the Hybrid and Energy-based approaches, the windings can be specified via the fill factor only.

7.7 Model Simplifications

Finally, it is possible to select certain simplifications of the model, in order to reduce the optimization execution time. To calculate core and winding losses, many harmonics of the current and voltage waveforms are considered: from the fundamental (i.e. mains) frequency up to the converter's switching frequency, and then several hundred more. By ignoring the high frequency (HF) effects, for DM waveforms, only the fundamental frequency harmonic is considered, and for CM waveforms, only the range from the fundamental to the switching frequency. Also, the calculation of proximity losses in the windings, which for EE cores involves a sometimes time-consuming mirroring procecire, can be ignored.

For CM inductors, the leakage inductance can also be ignored. Normally, as seen from the models in Section 2, the leakage inductance of each CM inductor design is calculated, simulated in the DM filter model, and then its waveforms extracted to be used in the calculation of the CM inductor's losses. If leakage is ignored, DM flux is not taken into consideration in the CM inductor loss calculations, and the current waveform of the DM inductor of that stage is used to estimate the winding losses, which greatly reduces the number of required DM filter simulations.

Be aware of course that each simplification, although it reduces the execution time, produces less accurate results. Whether a particular simplification is appropriate depends on the given design problem.

8. Parasitics and Additional Components

Once the inductor and capacitor design spaces have been specified, the final sub-tab of the Filter Design Parameters tab is for parasitic capacitances and additional components. This is shown in Figure 17.

8.1 Parasitic Capacitances

Here the values of the parasitic capacitances C_{eq} and C_g, in pF, are set. The influence and importance of these elements was explained in Sections 2.3 and 3.1. Therefore, be careful when setting these values and mindful of what they represent in your CM filter circuit simulation model.

8.2 Additional Components' Value Ranges

The additional components the names of which were entered in Model Set-Up tab are visible here. For each component, the value range it should take and the unit of that value can be specified. Additional components can be resistors, capacitors, or inductors (nothing else).

Please note that the additional components are NOT designed by the Optimizer, as is the case with the

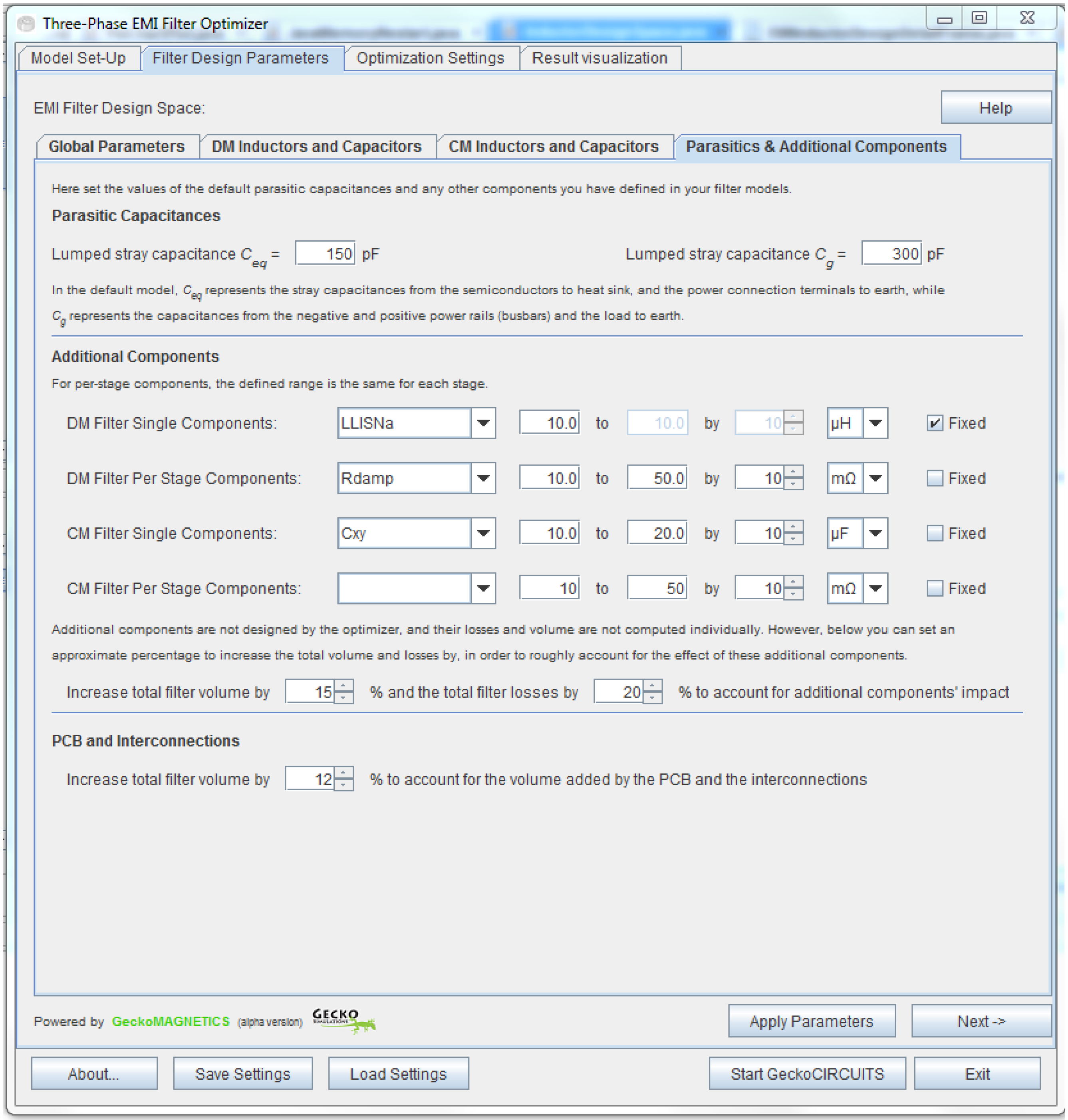


Figure 17: Optimizer screen for defining the values of paristic capacitances and user-defined additional components.

filtering components. Their losses and volumes are not calculated. They are left to the user to define because their presence may influence the losses of the filter inductors and capacitors, as well as the actual value chosen for the main filter components (i.e. the presence of an additional component may "move" the optimum design from one point of the design space to another).

Below the range definitions for the additional components, a volume and loss factor for them can be defined. The total calculated filter volume and losses will then be increased by the specified percentages to take into account the effect of the additional components. This is an approximation and the factors should be entered based on the additional component's projected size and possible values.

8.3 PCB and Interconnections

Finally, a factor for increasing the total volume of the EMI filter to account for the influence of the printed circuit board (PCB) and/or other component interconnections can be specified. This should ideally be entered based on the user's experience in building EMI filters.

The three additional factors are applied constantly to every valid design produced by the Optimizer, and therefore do not affect the outcome of the optimization. They are there only to provide potentially a more realistic estimate of the filter's total losses and volume.

9. Optimization Settings

The final tab used for defining an optimization problem is the "Optimization Settings" tab. It is shown in Figure 18. Here the simulation settings, optimization goal and approach, and output file is set.

9.1 Simulation Settings

For the simulation of the filter circuit models, the Optimizer needs to open two instances of GeckoCIRCUITS, in one of which the DM filter model being evaluated will be opened, and in the other the corresponding CM filter model. Here, the total simulation time for each model in each simulation run – expressed in the number of mains periods – has to be specified. It is important that steady-state is fully reached in the last period of simulation, as the last period is always used to extract filter waveforms. Also, as mentioned in Section 2, the CM and DM noise waveforms must be specified. They can be embedded in/generated directly by the model files, or they can be stored in a file and specified here. If the latter is the case, the Optimizer assumes that the filter models follow the general structure of those in Sections 2.2 and 2.3, that is, that the CM model has one voltage source generating the CM voltage waveform and that the DM model has three current sources generating the DM current waveform. For the DM waveforms, make sure that the three DM current waveforms are properly phase-shifted!

Also, the simulation time-step must be given. This should be chosen to provide the fastest possible simulation, but also to provide correct simulation results. This depends on the structure of your model and your application (e.g. converter switching frequency).

The Optmizer communicates with GeckoCIRCUITS over network ports. Therefore both it and GeckoCIRCUITS must have network access (despite the fact both are running on the same machine). If prompted by e.g. Windows Firewall to grant access to the Java platform, do so. If you cannot do this, contact your system administrator. The default ports should be free and open to use. If this is not the case, and you keep getting an error that you a connection to GeckoCIRCUITS cannot be established, try different port numbers.

9.2 Optimization Goals

The slider at the middle of the Optimization Settings tab defines an **optimization goal** between 0 and 1. A goal of 0 optimizer the filter for maximum power density, i.e. minimum volume, while a goal of 1.0 optimizes the filter for maximum efficiency, i.e. minimum losses. It therefore follows that a goal of 0.5 finds a half-way compromise between the maximum efficiency and maximum power density design. More precisely, once a pareto-optimal set of EMI filter designs is arrived at, a goal of 0.5 will cause the Optmizer to select a design which is equally distant (in terms of normalized losses and volume) from the minimum loss and minimum volume designs. Likewise, a design goal of 0.3 will chose a design closer the minimum volume design, while 0.7 will chose a design closer to the minimum loss design.

In order to calculate the efficiency of the filter, the total output power of the converter the filter is being designed for must be entered below the optimization goal. Also, the converter switching frequency must be entered, if not already specified in the Global Parameters tab.

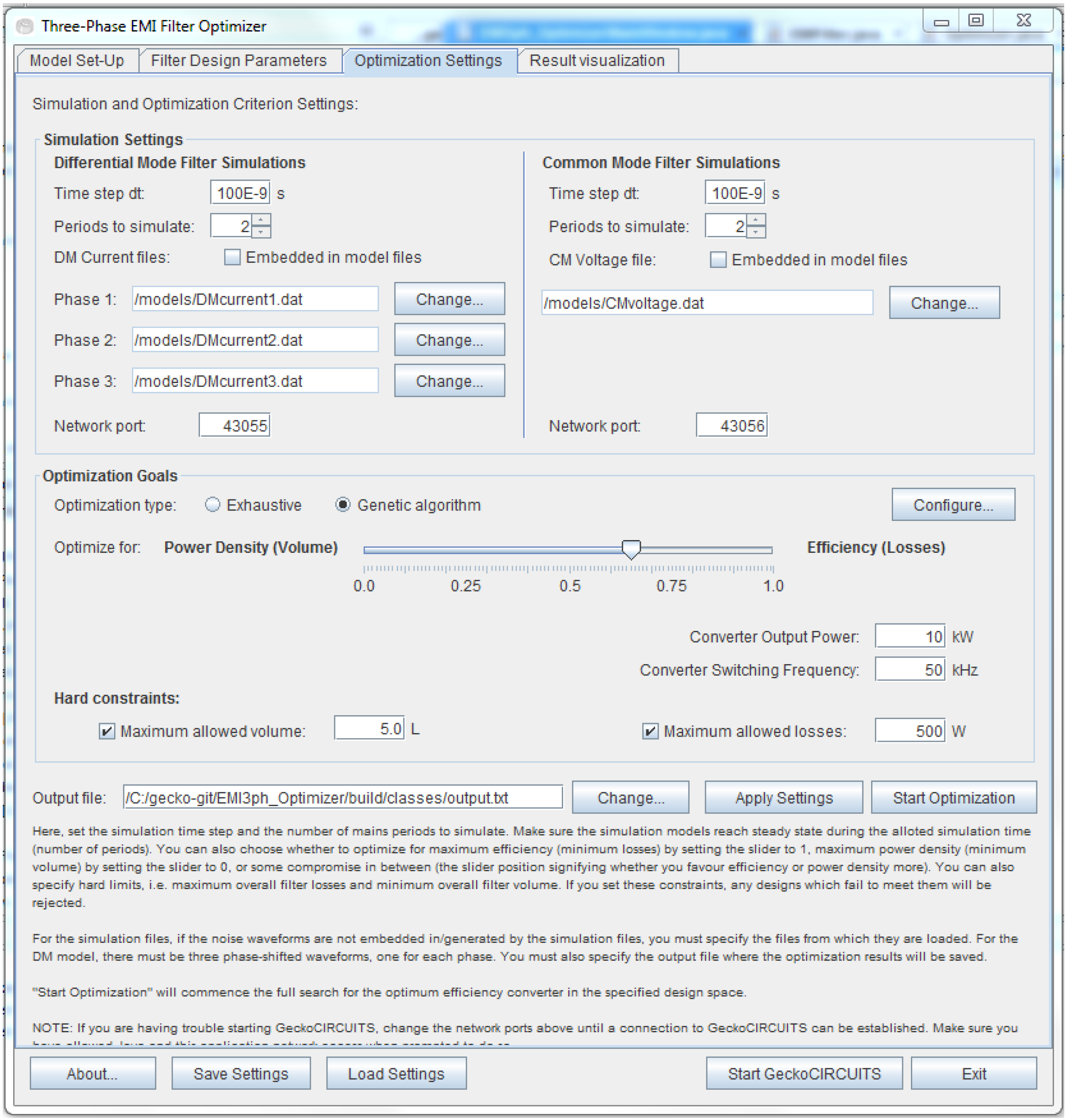


Figure 18: Optimization Settings

A set of hard constraints can also be defined, that is, the absolute maximum volume and losses allowed for the filter, so that any design which exceeds these constraints will be rejected.

The type of optimization must also be selected. Two are available: **exhaustive** (“brute-force”) and **genetic.** The exhaustive optimization, as it name suggests, evaluates every single possible design resulting from the defined overall filter design space. This is guaranteed to find the exact optimum each time, and to give repeatable results for a given design space. However, this can be extremely time-consuming. Many possible designs are dead-ends, and often the designs which take the longest the simulate (for example, those which overheat) are rejected. For this reason, optimization using genetic algorithms is also offered as an option, and should be used for all except the very smallest and most limited design spaces. However, be aware that genetic algorithms are stochastic in nature and are not guaranteed to always produce the same result. However, this will usually manifest itself in hardly noticeable manner – e.g., if the exact optimum for an inductor is a configuration with 1.5 mm air gap and 2.5 W losses, the genetic algorithm might produce as the best solution a configuration with 1.8 mm air gap and 2.7 W of losses.

The optimization type can be further configured by clicking the “Configure...” button on the right. This brings up the dialog show in Figure 19. In the first half of the dialog are the parallelization settings. These allow the calculations performed for the optimization to be split over several CPU cores on your computer, thereby shortening execution time. More precisely, inductor design evaluations are parallelized (nothing else). Therefore, if for each possible combination of filter LC values only one inductor design per stage is returned by the design space for evaluation, parallelization offers no speed-up. However, as this is usually not the case (tens, hundreds, or thousands of different designs need to be evaluated for each inductor), parallelization is usually beneficial, and the more processor cores available, the better.

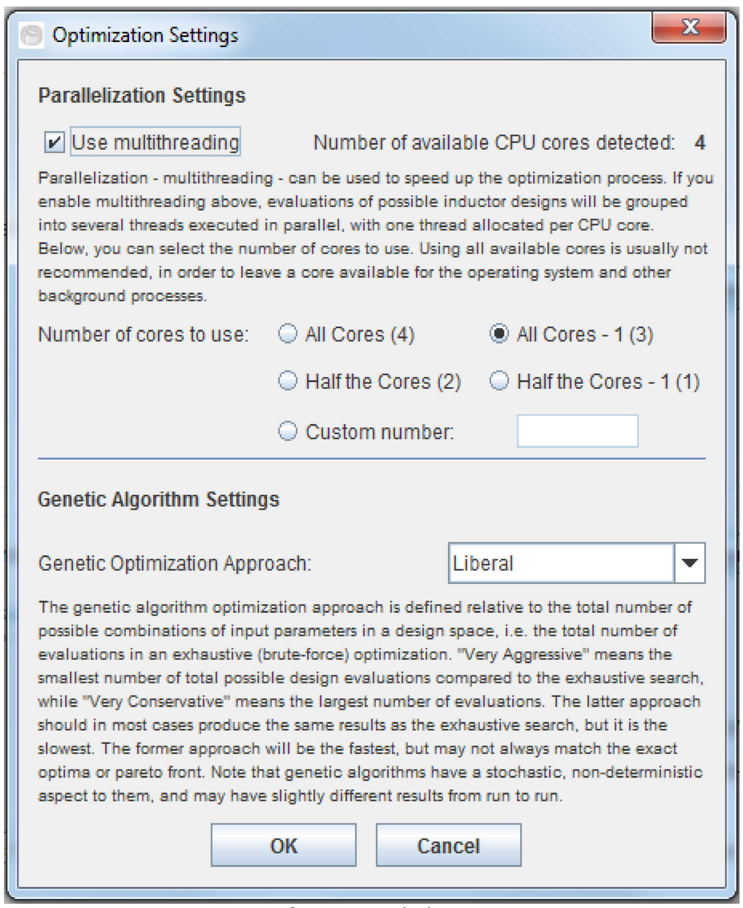


Figure 19: Optimization configuration dialog

The parallelization settings allow you to specify how many CPU cores to use. The setting “All Cores – 1” is default, and is recommended – one CPU core should be left free to handle the rest of your operating system running in the background. The Optimizer automatically detects the total number of CPU cores available. However, for Intel processors using “hyperthreading”, Java will detect double the number of actual physical cores. Therefore, if your processor uses hyperthreading, the recommended setting for parallelization is “Half the Cores” or “Half the Cores – 1”. It is also possible to enter a custom number of cores to use (e.g. 6 out a maximum 8).

The lower part of the optimization configuration dialog concerns genetic algorithms only. A **Genetic Optimization Approach** can be selected, ranging from "Very Aggressive" to "Very Conservative". This concerns the number of total design evaluations that the genetic optimization performs relative to the exhaustive optimization. "Very Aggressive" will try to find the optimum in the smallest number of evaluations, and therefore the shortest time, compared to the exhaustive optimization, while "Very Conservative" is the exact opposite. The more conservative the approach, the higher the chance that the exact optimum will be found. "Very Conservative" should be selected for small design spaces (e.g. less than 100 possible different inductor designs to evaluate per filter stage), while "Very Aggressive" should perform just as well as the others if the inductor design space is very large (e.g. hundreds of thousands, or millions of possible different inductor designs to evaluate per filter stage).

9.3 Output File

At the bottom, an output file should be specified. This is a file where the best filter design for each number of stages considered is written to in textual format, along with resutls (losses and volume). The file is written as soon as a given k-stage configuration is optimized, which means that it will potentially contain some data even if your optimization does not complete to the end (e.g. if your computer loses power, etc.). The output in this file is very simple, and is really meant just to be a backup of the results if you fail to save the full results once the optimization completes. Saving results is covered in Section 11.

10. Running an Optimization

Once all the parameters and settings are set, an optimization can be run. Before doing that however, please thoroughly read this section, as there are useful things to before starting an optimization, like saving the settings to a file.

10.1 Save and Load Settings to/from a File

The Optimizer provides you with the ability to save all the settings you have entered to a file, for later loading and re-use without having to enter everything manually again. To save the settings, click the "Save Settings" button at the bottom of the screen. The settings are saved in a text file in XML format, which given an *.e30 extension. You can open this file in a text editor and verify that all of your settings have been saved properly. You can later load these settings by clicking the "Load Settings" button at the bottom of the screen, and selecting the appropriate file. All of the settings entered, from all tabs, including inductor and capacitor design spaces, are saved.

10.2 Apply Names, Apply Parameters, Apply Settings

It is good practice to always click "Apply Names" when you are finished the Model Set-Up tab, "Apply Parameters" when finished wtih the Filter Design Parameters Tab, and "Apply Settings" when finished with the Optimization Settings tab. Or, alternatively, always move from tab to tab by clicking "Next", which will have automatically "press" these buttons when you move from one large category tab to the next.

When "Apply Parameters" is clicked, the Optimizer generates the possible LC value combinations for the filter design space. To indicate this, a dialog box appears with a message that this process is in progress. If this is taking very long, be aware that you have defined inductance ranges which are very wide and/or fine-grained, which has resulted in a large number of LC combinations, which will result most likely in a very long optimization run.

10.3 Sample Settings Provided with the Optimizer

A series of .e30 files is included with the optimizer, which you can load to immediately give the Optimizer a try with different settings. They are given in the "samples" sub-folder of the Optimizer package. It is recommended to first load and try the file "Fast1StageTest.e30". This optimizes a single-stage EMI filter –

it is **NOT** meant to be a realistic or feasible design, rather it is a collection of settings which will result in fast optimization and therefore allow you to explore quickly the full functionality of the program, especially regarding the results as described in Section 11. On most newer mid-to-higher-end machines, this optimization should complete in a few minutes.

The second sample set of settings included is “Sample2StageFilter.e30”. This is a more realistic design of a 2-stage EMI filter for a Swiss Rectifier converter with a load of 7.5 kW and a switching frequency of 36 kHz, and the resulting optimized design is something which is usable in a real application. Keep in mind however that it is also a constructed example primarily meant to show you the program features. On most newer mid-to-higher-end machines, this optimization should complete in about 40 minutes.

PDF report files resulting from these two sets of sample settings are also given in the “samples” folder. Note that these examples were created on a quad-core machine and are set to use 3 parallel processing threads. If your computer has less (or, preferrably, more) cores than this, adjust the appropriate setting as described in 9.2.

10.4 Starting the Optimization

It is recommended to close all other running programs before starting an optimization. The optimization process is computation and memory intensive, therefore consider your computer “dedicated” to the Optimizer when it is running. When everything is ready, pressing the “Start Optimization” button will open the Output Window shown in Figure 20, in which the optimization progress will be shown. When it is done, the Output Window will display a message saying, “Optimization finished.”

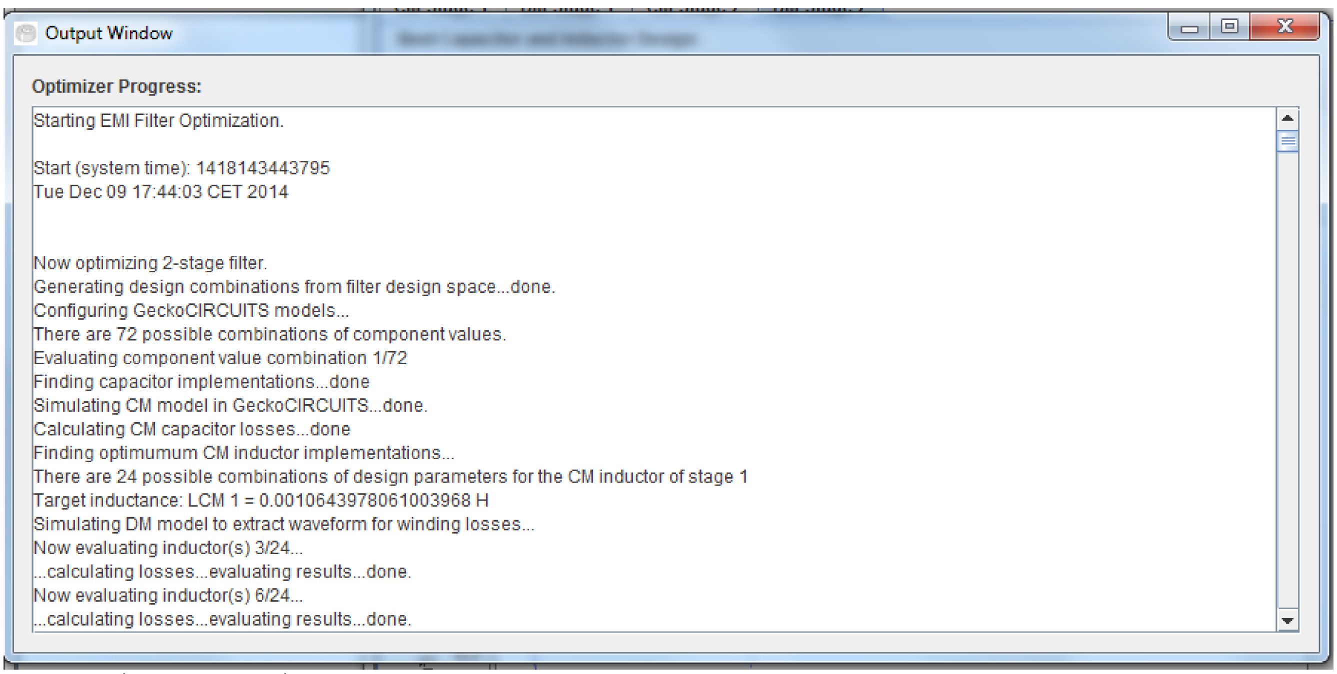


Figure 20: The Output Window

11. Optimization Results

When the optimization is finished, a message informing you of this will appear in the Output Window. If you then go tho the “Result Visualization” tab, you can view the outcome of the optimization run.

11.1 Best Filter Designs

In the first sub-tab of the Result visualization tab, you will see a list of best EMI filter designs. This is shown in Figure 21. The drop-down menu contains the best filter design, according to the set optimization goal, for each stage configuration considered. Therefore, if the Optimizer was set to consider 2-stage, 3-stage, and 4-stage filters, the drop-down menu will have three entries, the best (optimum) 2-stage, 3-stage, and 4-stage filter. Out of these, the overall best one according to the optimization goal (i.e., the overall optimum filter design) will be selected by default and its losses and volumes breakdown will be displayed in the graphs below the drop-down menu. By selecting different filter designs in the drop-down menu, the graphs will change to display the losses and volume of the selected filter.

All graphs (including all discussed in subsequent sections) can be **saved to a file** by right-clicking on them, then clicking on “Save As” in the menu which appears. Also, by right-clicking, they can be zoomed/unzoomed in different ways (see the options which appear after a right-click). The zoom level can also be changed by using the mouse scroll button.

11.2 Viewing Filter Details

When the “View Details” button in the lower right corner of the previously-mentioned sub-tab is clicked, this will open a window showing the details of the design of the currently selected filter. In this window, there is a tab for each DM and CM stage, as well as for the additional components (if any). This is shown in Figure 22. At the top of the tab for each stage, there is a text area which displays the detailed information about the capacitor and inductor designs used for that filter stage: for the capacitor, the components used, and the losses and volume; for the inductor, the core and wire used, number of turns, inductance (and leakage inductance for common-mode chokes), total volume, losses broken down by type, and the thermal properties.

The bar graph in the middle shows the losses of the filter stage, broken down into three categories: inductor winding losses, inductor core losses, and capacitor losses.

The third graph shows a pareto front plot of the inductor design space considered for this filter stage. All pareto-optimal designs resulting from the design space for the given operating conditions are shown, with the best one according the optimization goal, and therefore the selected one, marked in a different colour. This is the design for which the details are shown in the upper part of the screen. From this graph it can be seen how the optimization goal affects the inductor design which is selected. The inductor pareto front can be saved to a file by clicking the button below the graph. The graph data is saved in the *.csv format which can be easily imported into a spreadsheet program, e.g. Excel.

If the mouse is held over a particular point in the graph (i.e. an inductor design), a small text box will appear showing the basic information about that design.

In the additional components tab, the values of the additional components are shown, if any are present.

11.3 Viewing Inductor Details

The inductor designs can be examined in further detail by double-clicking on a point in the inductor pareto front graph. This will open the Inductor Design Details window, as seen in Figure 23. A loss breakdown of a single inductor is given by loss type in the first tab. In the second tab, a detailed description of the design is given. In the third tab, the current-dependent inductance characteristic is plotted. In the fourth and final tab, the inductor waveforms – flux density, voltage, and current – are

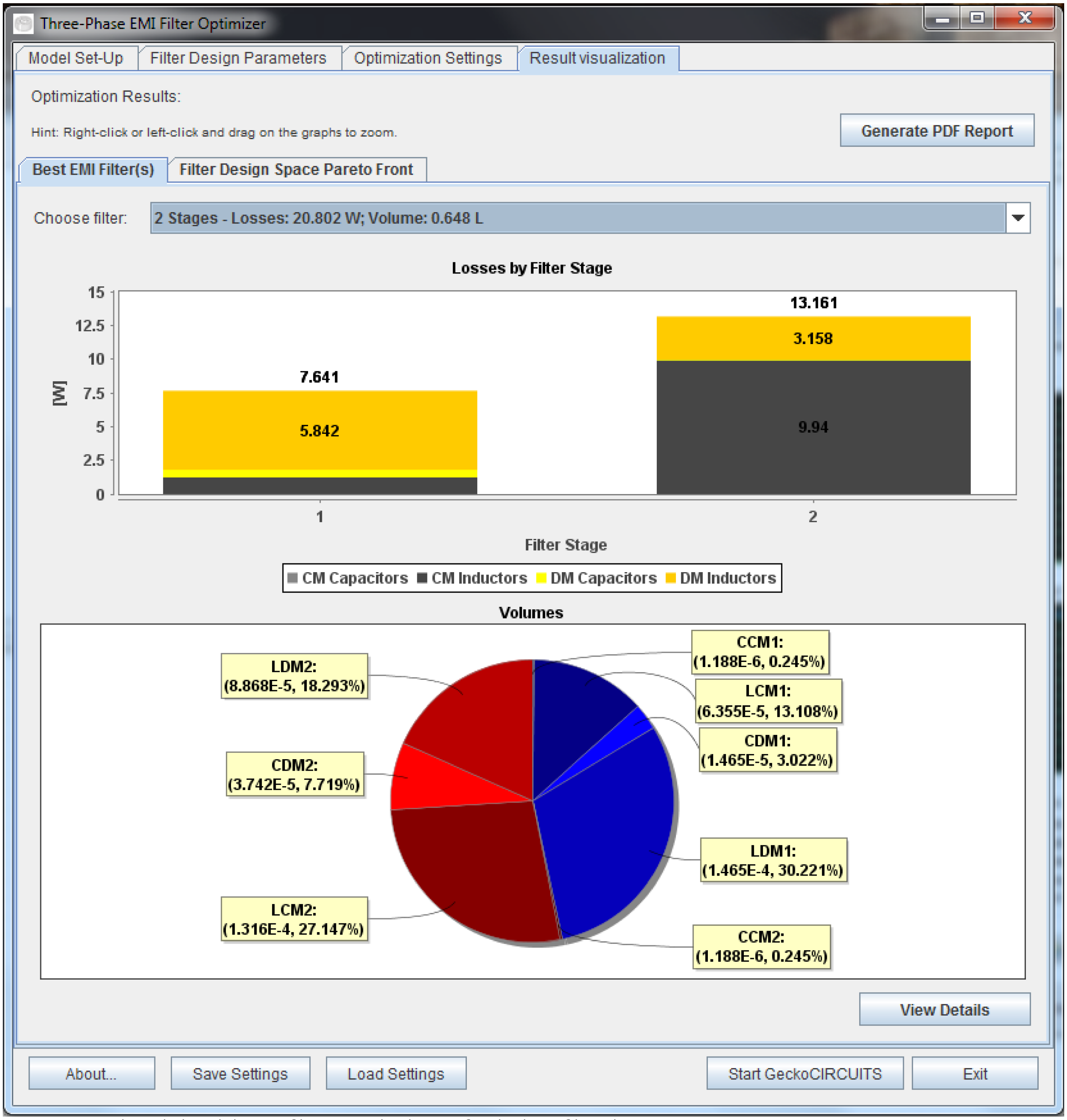


Figure 21: Results tab: breakdown of losses and volumes for the best filter designs.

shown. The waveforms can also be saved to file by clicking the appropriate button at the bottom of the screen. They are also saved in *.csv format (all waveforms are saved to one file, per inductor). Note that for CM inductors, if leakage inductance is neglected, only the CM waveforms will be shown. If it is taken into account, this window will display the total flux density waveform, and the DM and CM current waveforms – however, if you save the waveforms, the voltage waveform which is not displayed will also be saved to file.

Using the "Save Inductor Design" button in the lower right corner, the selected inductor design will be saved to a ***.gmd** file. This file can then be used to **load the inductor design into the commercial, full version of GeckoMAGNETICS** for further analysis. The saved waveforms can be loaded into

GeckoMAGNETICS by using the “Waveform From File” option in that program. It is expected that the loss calculations from the Optimizer and the full GeckoMAGNETICS version will be different, due to differences in the modelling approach, especially for CM chokes.

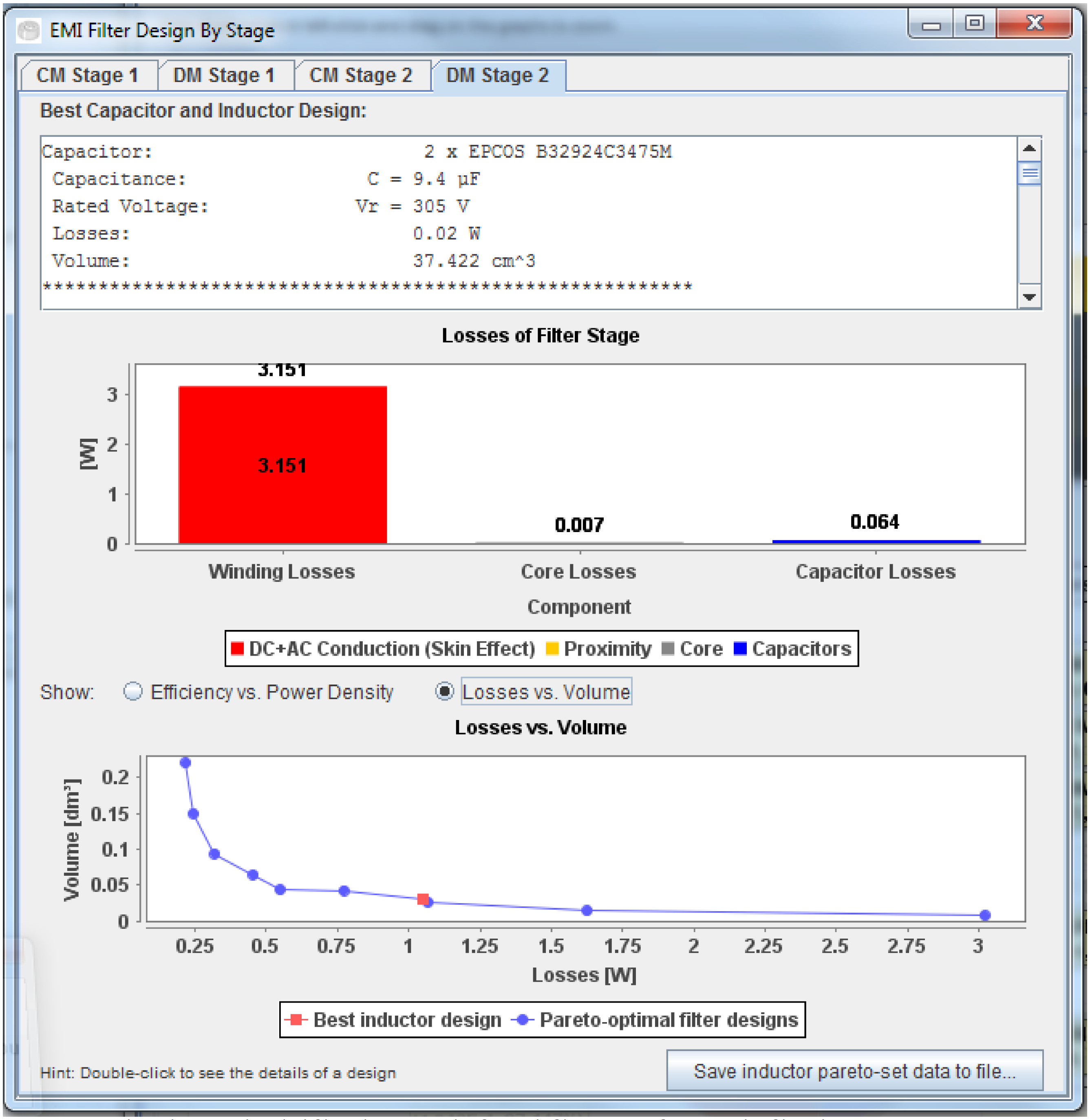


Figure 22: Window showing detailed filter design results for each filter stage of a particular filter design.

11.4 EMI Filter Pareto Front

Going back to the Result visualization tab of the main window of the Optimizer, the second sub-tab available is “Filter Design Space Pareto Front”, as shown in Figure 24. This shows, for each k-stage configuration of the filter, the set of pareto-optimal solutions, and the optimum filter selected in a different colour. This graph shows how the choice of optimization goal affects the selection of the filter

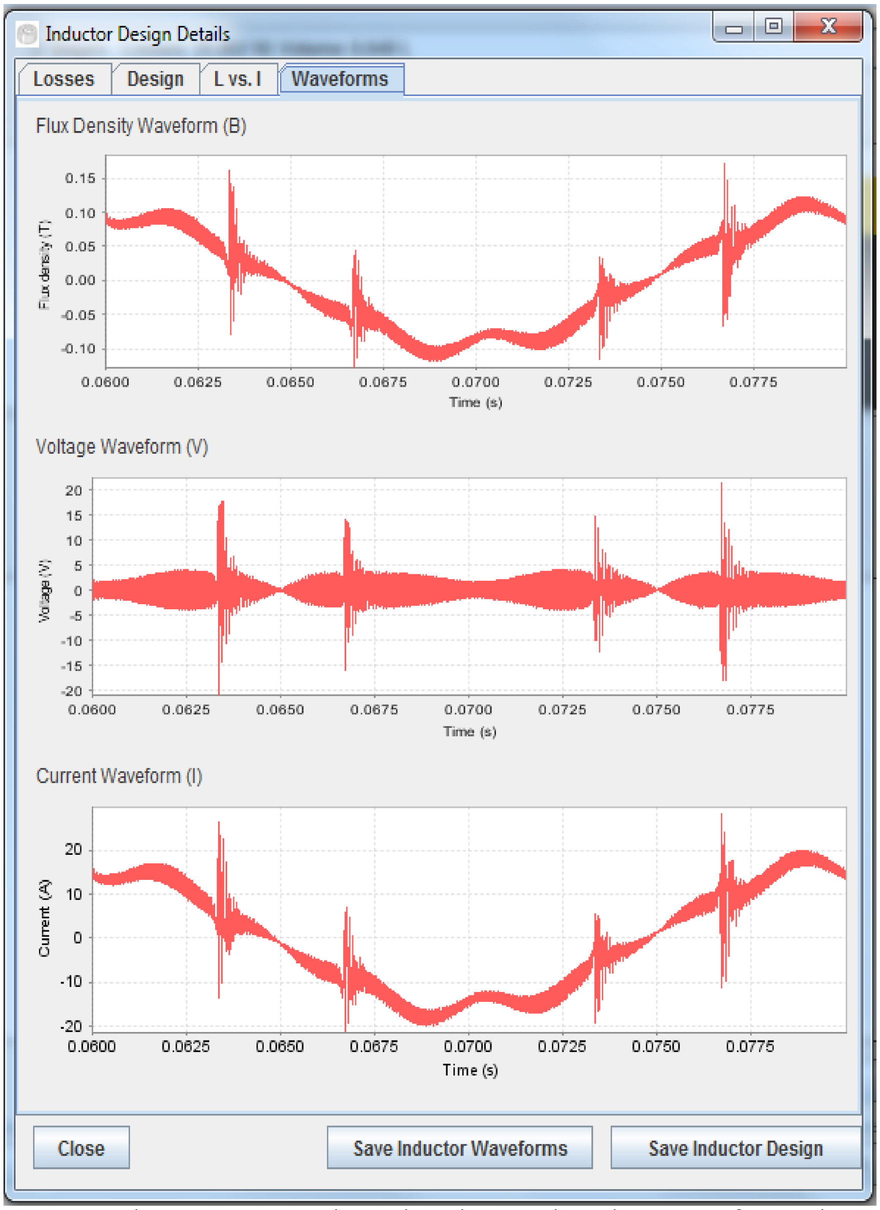


Figure 23: Inductor Design Details Window, showing the inductor waveforms tab.

design from the pareto front. This pareto front data can also be saved to a *.csv file by clicking the button below the graph.

Moving the mouse over a point in the graph, i.e. a filter design, will show a small text box with the main design parameters of that design. Double-clicking on on a particular point will open a new window showing the losses and volumes of that filter design, in the same format as in the “Best EMI Filter(s)” tab. From there on all the design aspects of the filter can be explored as described in the previous sections.

11.5 Generating a PDF Report

The Optimizer can generate a PDF report containing the optimization results. This can be done by clicking the “Generate PDF Report” button in the upper right corner of the Result visualization tab. The report will contain an overview of the defined filter design space and optimization settings, as well as of course the optimization results – the filter pareto front and the design details (with losses and volume) of the best filter at each k-stage configuration.

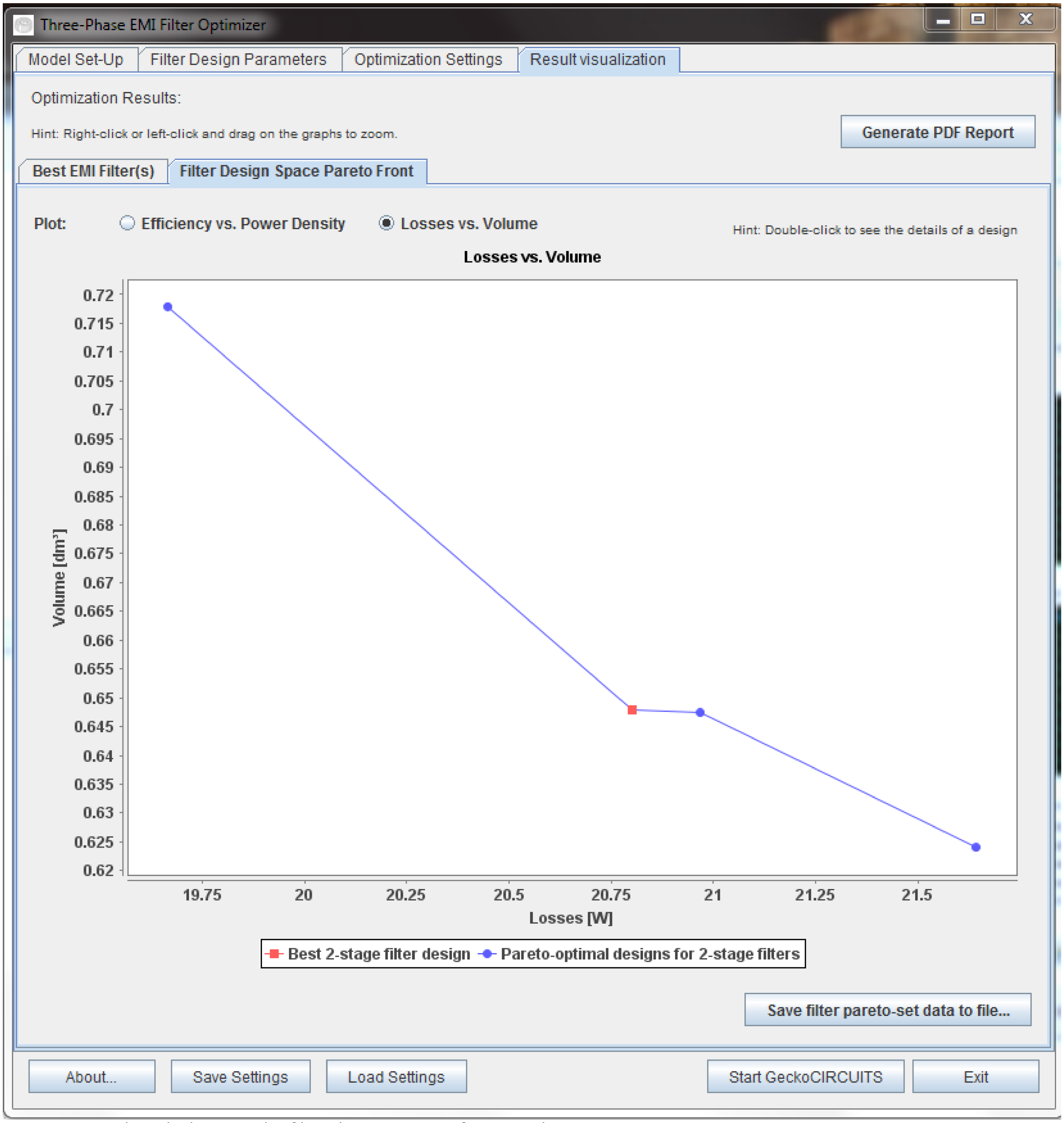


Figure 24: Results tab showing the filter designs pareto front graph.

Note that for the capacitor and inductor design spaces, one page is given per filter stage design space. This potentially means that for design spaces with a large set of selected predefined wires, cores, or capacitors, not all will be shown. The full input is always preserved when you save an *.e30 file of the Optimizer settings, and this file is human-readable.

Be aware that the PDF creation function places into the report the current state of the Optimizer input. So, for example, if you run an optimization, then change some input parameters, and then generate the PDF report, you will have a mismatch between the input parameters and the results. Therefore be careful to generate a PDF report immediately after the optimization is finished. Sample PDF reports are provided with the Optimizer for each of the sample settings files.

12. Troubleshooting

In this section, some common problems that may be encountered, and their solutions, if existing, are listed.

12.1 GeckoCIRCUITS starts but is blank

Problem: GeckoCIRCUITS starts, but only one window, not two, and no model file is opened, while the Optimizer displays an error message with the contents "Error communicating with GeckoCIRCUITS: null".

Solution: This is a time-out problem. GeckoCIRCUITS was too slow to start, and the connection on which the Optimizer was waiting timed-out. Usually, closing GeckoCIRCUITS and trying again will fix this problem. If it persists, contact the author of the Optimizer via the e-mail address listed on the first page.

12.2 Graphs of results not showing

Problem: There are blank white panels where graphs and plots of the results should be.

Solution: Move your mouse over the blank white panels, and the graphs will appear.

12.3 GeckoCIRCUITS will not start

Problem: GeckoCIRCUITS does not start at all, or it does start, but no models are opened, and no communication between the Optimizer and GeckoCIRCUITS can be established (error message thrown, etc.).

Solution: Probably you have not given GeckoCIRCUITS and/or the Optimizer (or Java in general) network access. Make sure you allow network access for Java, GeckoCIRCUITS, and the Optimizer program when prompted to do so by your operating system. Also, it could be that the ports the Optimizer is trying to use are occupied by another process or blocked on your system. Therefore, try using different port numbers (e.g. subtract or add 5, 10, 20, ... etc. from/to the default port numbers given). If the problem persists, contact your system adminstrator to adjust your network settings so that the Optimizer and GeckoCIRCUITS can run. If this still does not solve the problem, run the Optimizer from the command line (see Section 1.5), and send a description of your problem and any error messages that appear in the command line to the e-mail address on the first page.

12.4 Optimizer appears to be unable to find an additional component

Problem: In the Output Window, an error message such as "Parameter R does not exist in Element Cdamp1" appears.

Solution: This is not a problem. With additional components, the Optimizer dynamically attempts to determine whether your additional component is a resistor, capacitor, or inductor. It tries all possibilities until it gets one right – and if it "misses" on the first try, this error message comes back from GeckoCIRCUITS. However, if no other error messages or exceptions follow it, and the optimization continues normally, everything is fine and you need not worry.

12.5 Optimizer does not seem to evaluate all stated combinations of components

Problem: The optimizer states at the beginning of an optimal inductor search that there are e.g. 24 possible combinations of design parameters. However, it stops after evaluating e.g. 15, and goes on to the next inductor.

Solution: This is not a problem, but normal behaviour. The fact there are 24 possible combinations does not mean that they will all be evaluated. Firstly, if genetic optimization is used, it by design evaluates only a subset of the possible combinations (usually, at most up to 15%). Secondly, even if exhaustive optimization is used, some combinations are physically impossible (i.e. a given fill factor is unachievable, or a certain number of turns of a given wire cannot fit inside a given core) and therefore do not produce inductors for evaluation. Others do produce inductors which are immediately discarded because they are too large or because they saturate at the peak current (and such inductors are then never evaluated).

12.6 Strange messages printed to the Output Window when generating a PDF Report

Problem: When the "Generate PDF Report" button is clicked, strange messages starting with "INFO: About to return NULL.." appear in the OutputWindow several times.

Solution: You may ignore these messages, as long as they are not followed by any errors or exceptions. If there are errors and exceptions, and you find that your PDF file has not been saved after multiple attempts, please contact the author. If you exceptions and errors other than the one described above, but find that your PDF file has been generated and saved correctly, you can ignore them.

12.7 Optimization stops prematurely

Problem: The optimization process is interrupted and stops before it is finished, and various error messages and Java exceptions are printed to the Output Window.

Solution: Please save the settings you used when this problem occurred, copy all the error messages into a text file, and send both with a complete description of your problem to the author using one of the e-mail addresses on the first page.

12.8 Maximum value of parameter range not reached

Problem: An e.g. additional component range is set to be from e.g 10 to 30 mΩ by 2 mΩ, but the maximum value ever used by the optimizer is 28 mΩ.

Solution: This can occur due to Java floating-point number inprecision. The Optimizer usually accounts for such problems, but it is possible that this problem sneaked in somewhere in the code. To solve the maximum of the range to be less than one increment greater than your desired actual maximum vale. Using the above example, set the range to be 10 to 31 by 2. In this case, the desired actual maximum of 30 is guaranteed to be reached.

12.9 Genetic optimization taking longer than exhaustive optimization

Problem: For the same set of input parameters, the genetic optimization takes longer than the exhaustive optimization.

Solution: A genetic algorithms needs a certain minimum number of iterations to produce results. The above problem occurs for small design spaces, where there are less than 50 possible inductor designs per stage to be evaluated. In such cases, it is better to simply use an exhaustive optimization approach.

12.10 Optimization is taking too long

Problem: The Optimizer generates too many LC value combinations, and the optimization is taking days to complete.

Solution: Be aware that every single LC value combination will be evaluted by the optimizer, regardless of the optimization type and approach. To generate a small set of LC combinations, set all stages to have equal attenuation, and then fix the inductance at each stage to an unrealistically low value (e.g. 1 µH). This will cause the Optimizer to reject your input, and generate for each stage, three combinations – with the capacitance of that stage at 25%, 50%, and 100% of the maximum allowed per-stage capacitance. For a 2-stage filter, taking into account both the CM and DM stages, this will generate 81 LC combinations to evaluate.

13. Note on Common-Mode Inductor Leakage Inductance Calculation

One important aspect of the Optimizer is the calculation of the leakage inductance of the three-phase common mode inductors. This is done in two ways, depending on the data available. For custom cores (i.e. those defined by the user directly, and not entered in the database), the leakage inductance is calculated based on an empirical equation which takes into account the core geometry[8]. However, it has been shown[9] that this approach can yield inaccurate results. A better, but more time-consuming approach is to perform 3D simulations of cores and to extract for each one an appropriate leakage inductance A_L value[10]. For the VAC cores supplied by default in the Optimizer's database, such leakage data is provided[11]. The Optimizer will **ALWAYS** use leakage A_L data to calculate the leakage inductance of a common-mode inductor, IF such data is present for a given predefined core in the database. If there is no such data, the less accurate empirical equation will be used instead.

Note, however, that if you are using cores for which leakage data exists, that you are using these cores in manner which produces meaningful results. The leakage data is not tied to a specific core only, but also to a specific core material. For the VAC cores in the database, the same basic material – VITROPERM 500F – is used, but with slightly different properties from core to core. The only variant of VITROPERM which is given in the default database is VITROPERM 500F-18k, and it does **NOT** correspond to the material used in all of the cores in the database. The VAC cores in the database which use VITROPERM 500F-18k are marked with a “ **+** ” at the end of their names. These cores, and their leakage data, can be safely used with VITROPERM 500F-18k. Another set of cores which uses a variant of the material very similar to VITROPERM 500F-18k, and for which it can be an acceptable approximation, are marked with “ **#** “ at the end of their names. For the remainder of the VAC cores, if you wish to use their leakage data meaningfully, please **first enter into the database** the characteristics of the appropriate VITROPERM 500F variant.

---

8 M. L. Heldwein, L. Dalessandro, J. W. Kolar, “The Three-Phase Common-Mode Inductor: Modeling and Design Issues”, IEEE Transactions on Industrial Electronics, Vol. 58, No. 8, pp. 3264-3274, August 2011.

9 D. O. Boillat, J. W. Kolar, J. Mühlethaler, “Volume Minimization of the Main DM/CM EMI Filter Stage of a Bidirectional Three-Phase Three-Level PWM Rectifier System”, Proceedings of the IEEE Energy Conversion Congress and Exposition (ECCE USA 2013), Denver, Colorado, USA, September 15-19, 2013.

10 Ibid. See: http://www.pes.ee.ethz.ch/uploads/tx_ethpublications/24_Volume_Minimization_ECCE_USA_Boillat.pdf

11 This data has been provided by D. O. Boillat and was derived from simulations performed for the purpose of the subject of investigation of his paper, which is is quoted in the previous two footnotes.

If you are ignoring leakage inductance in your optimization, then the above considerations will have no effect on your actual optimization results. However, in the design details, a calculated leakage inductance for each CM choke will still be displayed, and in the interpretation of the number, the above still applies.

CM leakage data for a core can be viewed, deleted, or modifying by clicking the "CM Leakage Data" button in GeckoDB (see Figure 16). This opens the leakage data dialog as seen in Figure 25. The A_L value is given for different numbers of the given core stacked. You can import this data into GeckoDB from a text file constructed in the same format.

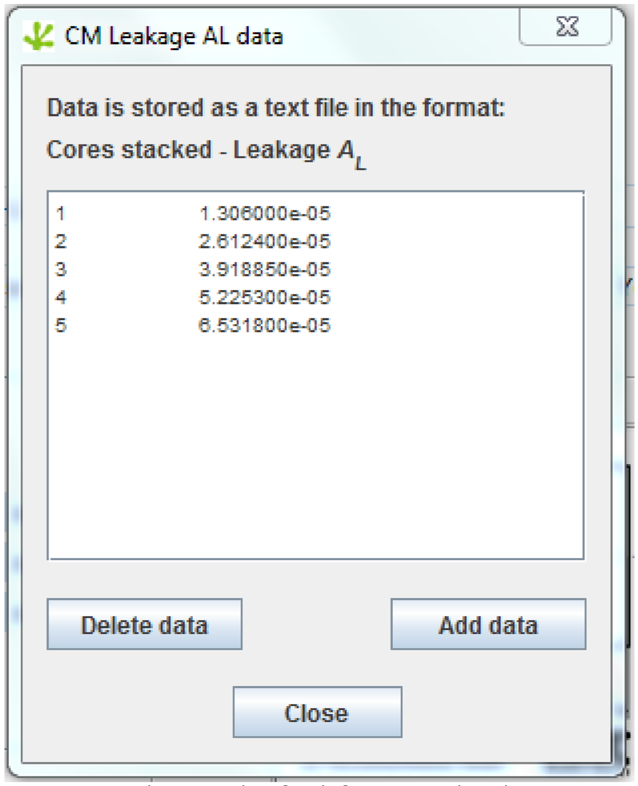


Figure 25: GeckoDB Dialog for defining simulated or measured leakage inductance values for a core.

Document version: December 10th, 2014

Andrija Stupar