



PLECS

*Tutorial*

## Analysis Tools in PLECS Standalone

[www.plexim.com](http://www.plexim.com)

- ▶ Request a PLECS trial license
- ▶ Check the PLECS documentation

---

# Contents

<b>1 Introduction</b>	<b>2</b>
<b>2 Tools Overview</b>	<b>2</b>
2.1 Steady-State Analysis . . . . .	2
2.2 Small-Signal Analysis . . . . .	6
<b>3 Exercises</b>	<b>8</b>
3.1 Steady-State Analysis . . . . .	8
3.2 Small-Signal Analysis . . . . .	9

# 1 Introduction

The PLECS Analysis Tools can be used to perform steady-state and small-signal analyses. The Steady-State Analysis tool can be used to find the steady-state operating point of the system and the Small-Signal Analysis tools can be used to obtain open-loop and closed-loop transfer functions. Three types of small-signal analyses are available: AC Sweep, Impulse Response Analysis and Multitone Analysis.

In this tutorial you will learn when and how to use these tools. The tutorial does *not* explain the theoretical background of each analysis tool but focuses on its application in a system-level simulation. To learn more about the theoretical background of each technique, please refer to the PLECS User Manual [1].

**Before you begin** Ensure that you have the reference files that you can compare with your own models at each stage of the exercise.

## 2 Tools Overview

PLECS Standalone offers special steady-state and small-signal analysis tools that can be accessed from the **Simulation + Analysis tools...** menu. The following sections give a quick overview on the most important properties of the different tools.

### 2.1 Steady-State Analysis

Most dynamic systems undergo an initial dynamic transient process and eventually settle to a steady state. The steady state of a system is reached after a time period  $t_s$  when all the state variables  $x$  either don't change at all

$$\text{Constant (DC) Operating Point: } \dot{x}(t) = 0, \forall t > t_s \quad (1)$$

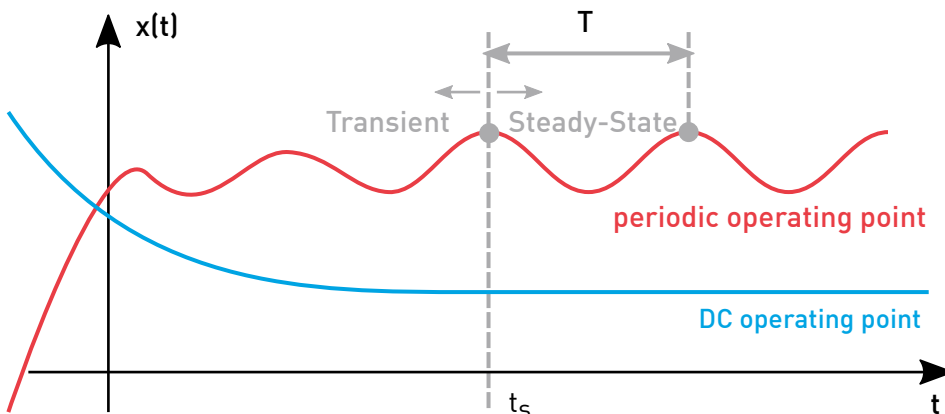
or change in a periodic manner

$$\text{Periodic Operating Point: } x(t+T) = x(t), \forall t > t_s. \quad (2)$$

The two cases are depicted in Fig. 1. These are also the two options to choose from in the Steady-State Analysis tool parameter window.



**Note:** When performing a steady-state analysis, only the steady-state value of the state variables  $x$  can be calculated, but not the time  $t_s$  when the steady state is actually reached.



**Figure 1: Periodic and DC operating points**

## Algorithm

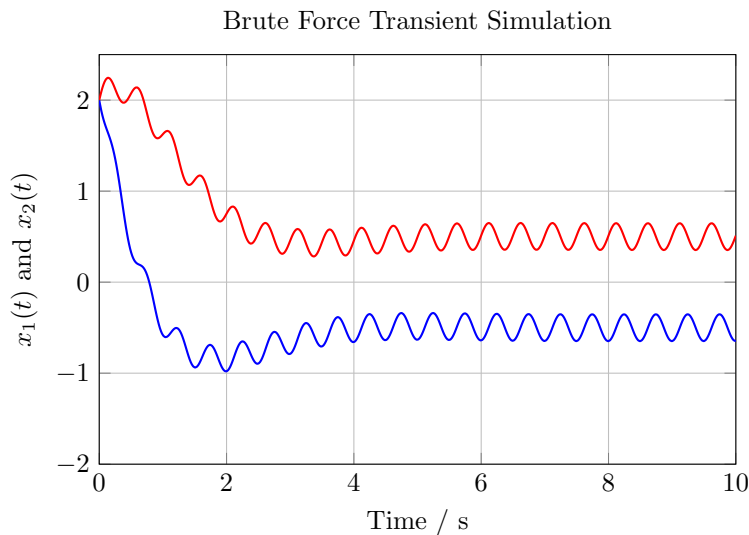
**Constant Operating Point** The steady state of a system with a constant operating point, i.e. a passive filter, can be calculated directly from the state-space description without a transient simulation. Using the property of a constant operation point in Eq. 1 the steady state is calculated as

$$\dot{\mathbf{x}} = 0 = A\mathbf{x} + B\mathbf{u} \Rightarrow \mathbf{x} = A^{-1}B\mathbf{u}. \quad (3)$$

**Periodic Operating Point** The Steady-State Analysis tool for a periodic operating point performs the calculation in the phase space of the system. A concrete example is necessary to be able to visualize this process. Consider the following dynamic system:

$$\dot{\mathbf{x}} = \underbrace{\begin{pmatrix} -1 & -1 \\ 1 & -1 \end{pmatrix}}_A \mathbf{x} + \underbrace{\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}}_B \underbrace{\begin{pmatrix} \sin(4\pi t) \\ \cos(4\pi t) + 0.5 \end{pmatrix}}_u \quad \text{with} \quad \mathbf{x}(0) = \begin{pmatrix} 2 \\ 2 \end{pmatrix}. \quad (4)$$

One can solve the differential equation by simulating Eq. (4) over the time axis until the point  $t_s$ . However, such a simulation might take a long time because of a slow initial transient behavior of the system. The result for the example in Eq. 4 is given in Fig. 2. A PLECS model `steady_state_analysis_example.plec` for this example is distributed together with the solutions for the exercises in this tutorial.



**Figure 2: Transient solution of example model**

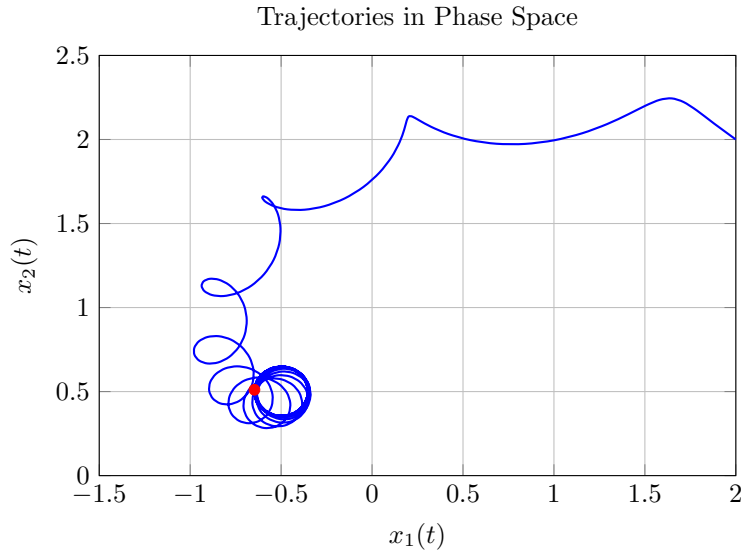
Another approach to calculate the steady state is to determine the periodicity of the system by looking periodic sources. In general the system period is calculated as the least common multiple of all AC periods in the system. In this example there are two AC terms in the input vector of the state-space description in Eq. (4). The Sine and Cosine function have a periodicity of 2 Hz. In steady state the trajectories in phase space ( $x_2(t)$  vs.  $x_1(t)$ ) show this periodicity as a circle, see Fig. 3.

Due the periodicity of the solution the following equation holds in steady-state

$$\mathbf{x} = \mathbf{F}_T(\mathbf{x}). \quad (5)$$

Where  $\mathbf{F}_T$  denotes a transient simulation for one system period  $T$  starting from a state  $\mathbf{x}$ . One can rearrange the equation to get

$$\mathbf{f}(\mathbf{x}) = \mathbf{F}_T(\mathbf{x}) - \mathbf{x} = (0, 0). \quad (6)$$



**Figure 3: Trajectories of example model. The red dot marks the steady-state value.**

The function  $f(x)$  is an auxiliary function whose root coincides with the steady state of the system. Thus, the calculation of the function  $F_T(x)$  allows to effectively transform the periodic system in system with constant operating point.

### Visualization

Finding the steady state in phase space can be visualized with the vector field in Fig. 4. Every point in phase space  $(x_1, x_2)$  is assigned a vector  $f(x) = (f_1(x_1, x_2), f_2(x_1, x_2))$  in blue color. The start point of the newton iteration is at  $(2, 2)$  where the arrows are long and therefore far away from steady state.

The red line shows an subsequent transient calculations of  $F_T(x)$  with the start point at  $(2, 2)$  and the final steady-state value at  $(-0.65, 0.5)$ . This calculation is time consuming. At the final point the arrows have zero length which means the auxiliary function is zero and therefore steady-state is reached.

On the other hand calculating the steady-state using Eq. 6 and a Newton iteration is much faster than a transient simulation. In this example the iteration converges already in the first step. The result is shown in Fig. 4. There are only two points, one is the start point and one is the final steady state.

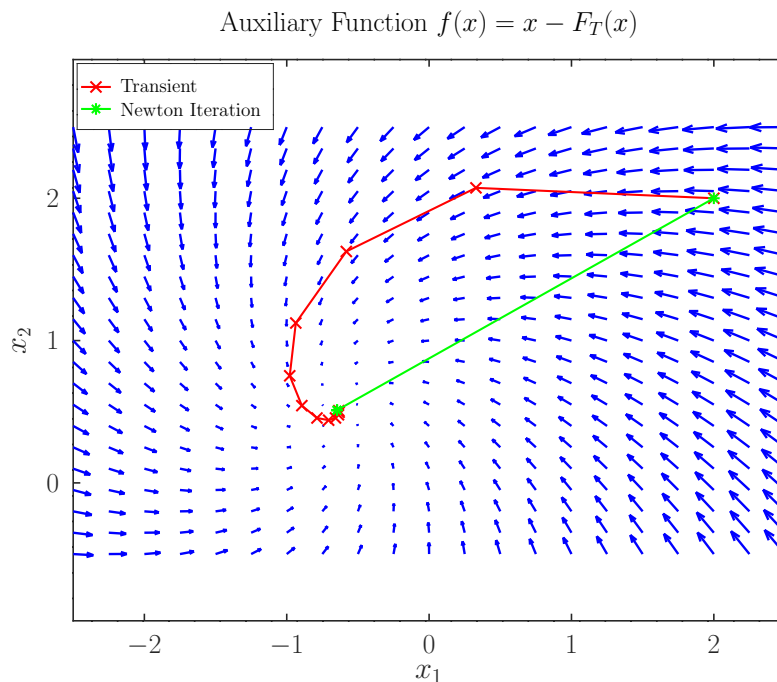


**Note:** In the general case the difference in calculation speed between transient and iterative calculation might not be as clear as in this case. Especially for more complex power electronic systems with closed-loop control and a switched power converter the Newton iteration typically takes more steps to converge.

### Requirements

Several model/simulation requirements have to be met before running a steady-state analysis:

- The system period  $T$  is the least common multiple of the periods of all AC sources in the system. If auto is selected in the **System period** parameter field, PLECS will try to calculate this number. However, if special blocks with an implementation unknown to PLECS are used, the calculation might be wrong and the system period needs to be calculated by the user. If the calculated parameter does not reflect the true system period or an integer multiple of it, the analysis will yield meaningless results or fail to converge altogether.



**Figure 4: Vector field of the auxiliary function  $f(x)$**

- The system period  $T$  has to be known *prior* to running an analysis and specified in the **System period** parameter field. For example, in case of converters with variable switching frequency control or variable speed drives the steady-state analysis can not be used.
- There must be no transient(s) in the system during the analysis except for periodic sources, e.g., do not add a switched/stepped load that will change the system impedance matrix sometime during a transient simulation. Otherwise the steady-state analysis will not converge.



**Note:** If these requirements are met but the steady-state analysis does not converge it can be useful to increase the **number of init. cycles** in the **Options** tab of the steady-state analysis tool parameter window. This performs a brute-force initial transient simulation to bring the system states a bit closer to the steady state. This can improve the convergence of the tool.

## Electro-Thermal Analysis

Finding the steady-state operating point of an electro-thermal model can be difficult because the thermal system typically has transients with significantly larger time constants than the electrical system. There are different approaches to tackle this problem:

- 1 Use the Steady-State Analysis Tool
  - + Fast calculation of the steady-state operating point of electrical and thermal states
  - Calculation bound to convergence of Newton iteration
- 2 Perform a transient simulation
  - + Robust
  - Might take a very long time to finish, require excessive data storage
- 3 Artificially decrease thermal capacitances and perform a transient simulation
  - + Will decrease the thermal time constant (faster simulation)

- Will increase the temperature ripple (and possibly make the system thermally unstable)

**4 Use separate electrical and thermal simulations**

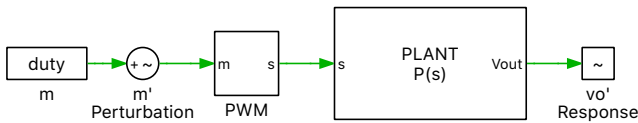
- + Robust
- Needs two simulation models and simulations, doesn't allow temperature-dependent resistors

**2.2 Small-Signal Analysis**

PLECS Small-Signal Analysis can be used in a controller design workflow.

**Control-to-output (plant) transfer function**

The first step in the design process is to find the transfer function of the system, i.e. the plant. The transfer function is a measure of how much and how fast the output changes in response to input changes. In the context of power conversion, the plant transfer function describes the influence of input disturbances in an unregulated converter to the load.



**Figure 5: System with an open-loop controller**

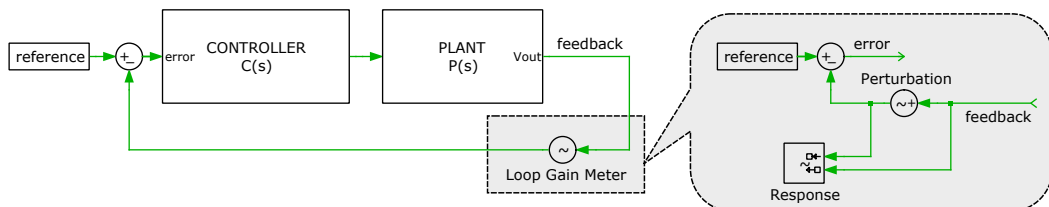
For example, the control-to-output (plant) transfer function of the system in Fig. 5 is:

$$P(s) = \frac{V_o(s)}{m(s)} \tag{7}$$

Small-Signal Analysis components from the PLECS Control library are used to obtain this transfer function. The “Small Signal Perturbation” block generates (or injects) the appropriate perturbation signal and the “Small Signal Response” block measures the system response for the Small-Signal Analysis.

**Open-loop gain**

Once the controller is designed based on the plant transfer function, the stability of the closed-loop system can be predicted by calculating the open-loop gain. The “Small Signal Gain” (or Loop Gain Meter) block, shown in Fig. 6, measures the open-loop gain of a closed control loop. This block first uses the “Small Signal Perturbation” block to inject a perturbation into a feedback loop and then the “Small Signal Response” block to measure the system response. This allows the “Loop Gain Meter” component to cut off the feedback loop, while still updating the values of the feedback output to calculate the loop-gain. The measured open-loop gain of the closed-loop controlled system is  $C(s) \cdot P(s)$ .



**Figure 6: System with a closed-loop controller**

Then, if the gain and phase margins, which determine the stability, are met, the design can be verified by simulation. Otherwise a design iteration can be performed.

## Overview of the different types of small-signal analyses

There are different analysis tools available to calculate the small signal response of the converter. Similar to performing a system linearization by hand, the tools first calculate the steady-state operating point (steady-state analysis) of the system and then apply a small perturbation. It is important to use the correct tool for a given problem. The table below provides an overview of the main properties of each tool.

**Table 1: Analysis tools overview**

Tool	# Simulations per Analysis	Loop Gain Calculation	Steady-State Analysis	Type of Steady-State
Multitone	2	Yes	Transient	Periodic & DC
AC Sweep	n	Yes	Iterative	Periodic & DC
Impulse Response	1	No	Iterative	Periodic

Some of the main conclusions that can be drawn from Tab. 1 are:

**# Simulations per Analysis** A frequency response consists of  $n$  individual frequencies. Depending on the type of perturbation (sinusoidal, Dirac pulse, etc.) different numbers of simulations are needed. The analysis will take longer as more simulations are needed.

**Loop Gain Calculation** The Impulse Response Analysis does not support loop gain calculation, but only should be used to determine the control-to-output (plant) transfer functions (or to measure output impedance).

**Steady-State Analysis** Only the Multitone Analysis uses a “brute force” transient analysis to reach steady-state conditions. This technique is required for systems where the system period is *not* known before the analysis, such as a resonant converter.

**Type of Steady-State** Both the Multitone and AC Sweep Analyses can analyze the small-signal behavior of a system with a constant (DC) operating point, such as passive filters.



## 3 Exercises

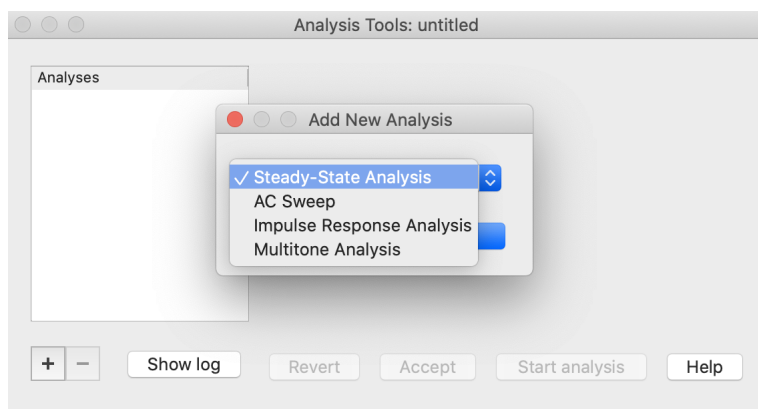
### 3.1 Steady-State Analysis

In this tutorial we will look at performing the Steady-State Analysis for a periodic operating point, with the help of a single-phase inverter with thermal model.



#### Your Task:

Locate and open the PLECS model `analysis_tools_steadystate_start.plecs` from the reference files. Start the simulation and open the scopes labeled “Electrical” and “Thermal” to observe the simulation results. You will observe that the transients of the electrical system settle out much quicker than the thermal system.



**Figure 7: Analysis Tools dialog**

Now we will set up the Steady-State Analysis for the model.



#### Your Task:

- 1 Open the Analysis tools dialog box from the **Simulation + Analysis tools...** menu. Click on the **+** icon on the lower-left side to add a new analysis and choose **Steady-State Analysis** from the drop-down menu, as shown in Fig 7. Because the model features two periodic sources (the Sine Wave and Symmetrical PWM blocks) set the **Operating point** to periodic in the **Setup** tab.
  - ❓ What is the system period of the model? Will the option auto work in this example?
  - A** The Sine Wave block is configured at 60 Hz ( $T_{s1} = 0.01666667\text{s}$ ) and the switching frequency is 50 kHz ( $T_{s2} = 20\mu\text{s}$ ). The least common multiple of these two periods is 0.05 s. The same result is calculated by PLECS with the auto option for the **System period** parameter.
- 2 Leave the **Simulation start time** parameter as 0. The desired start time must be set beyond any timed events such as step events, ramps, or other system disturbances.
- 3 Then set the **Show final cycles** parameter to 5. This parameter defines how many steady-state cycles are shown upon the completion of the analysis.
- 4 The Steady-State Analysis is based on the Newton Raphson technique, and the parameters in the **Options** tab help fine-tune the algorithm. A detailed explanation is provided in the PLECS User manual [1]. For now, leave these parameters as their default

values, then click on **Accept** and **Start analysis** to begin the Steady-State Analysis. Note that you can see the progress of the analysis by clicking the **Show Log** button.

**?** What can you do if the analysis *does not converge*?

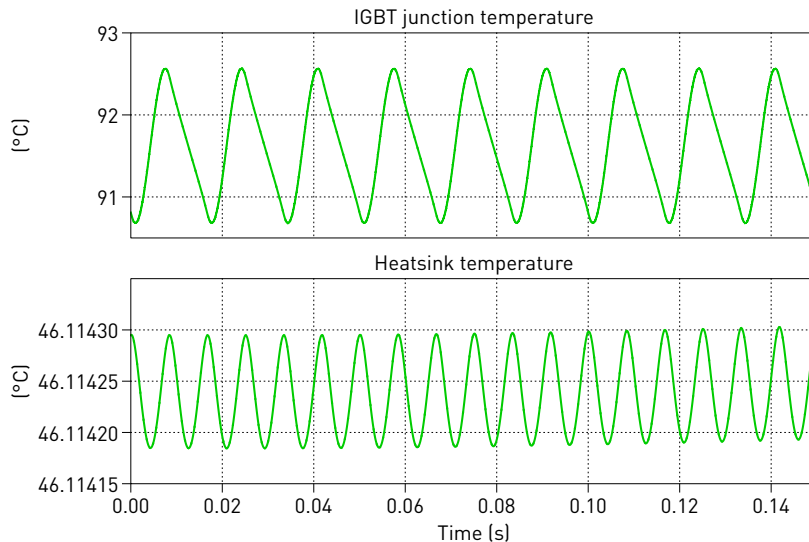


- First, verify if the system period is defined and calculated properly.
- Next, convergence could be improved by increasing the **Number of init. cycles** in the **Options** tab. This parameter makes a brute force initial simulation to bring the system closer to a steady state. This can help improve the convergence of the tool but it takes PLECS longer to perform the overall steady-state simulation. Note that the length of one init. cycle is equal to one system period.

Increasing the **Max. number of iterations** rarely helps as the Newton iteration shows quadratic convergence (assuming that it actually converges).



At this stage, your model should be the same as the reference model, `analysis_tools_steadystate_1.plecs`. Your simulation results obtained should be the same as those shown in Fig. 8 (thermal simulation results).



**Figure 8: Simulation results from the Steady-State Analysis**

## 3.2 Small-Signal Analysis

### Calculate the control-to-output transfer function

In this section, let's calculate the duty-ratio-to-output voltage transfer function of the buck converter shown in Fig. 9 using the PLECS Analysis Tools.

An approximate plant transfer function of a buck converter is:

$$P(s) = \frac{V_o(s)}{m(s)} = \frac{V_{in}}{1 + s \frac{L}{R} + s^2 LC}, \quad (8)$$

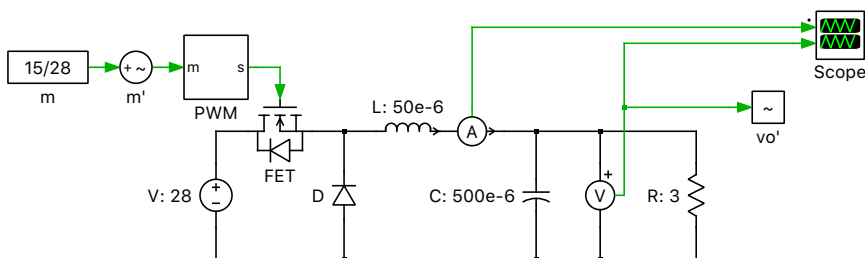
where  $m(s)$  is the duty ratio,  $V_{in}$  is the input voltage,  $R$  is the load resistance,  $C$  is the output capacitance and  $L$  is the inductance.

**Your Task:**

- 1 Add Small-Signal Analysis components to the PLECS model labeled `analysis_tools_openloop_start.plecs` from the reference files. This is a model of a buck converter operated at a switching frequency of 100 kHz.
- 2 Once you locate and open the PLECS model, drag and drop the “Small Signal Perturbation” block onto the schematic, and place it in between the duty ratio (constant block labeled “m”) and the “PWM” block.
- 3 Then set the show feed-through input parameter to on. Next, drag and drop the “Small Signal Response” block into the schematic and connect its input terminal to the output of the voltmeter measuring the load voltage. Leave all the parameters as their default values.



At this stage, your model should be the same as the reference model, `analysis_tools_openloop_1.plecs`. Your schematic should be as the one shown in Fig. 9.



**Figure 9: Schematic of the buck converter with small signal blocks**

**?** As explained in Tab. 1, there are three types of PLECS Analysis tools that can be used to obtain the open-loop transfer function: AC Sweep, Impulse Response and Multitone Analysis. Which analysis should be used for this model?

**A** For a periodic system like the one shown in Fig. 9, the Impulse Response method should be used to determine the open-loop transfer function because it's faster and computationally less expensive than the AC Sweep.



**Your Task:** Perform the Impulse Response Analysis of the model.

- 1 First, run a Steady-State Analysis as described in section 3.1, to ensure it converges.
- 2 Then, click on the + icon on the lower-left side of the dialog and choose Impulse Response Analysis from the drop-down menu.
- 3 From the **Setup** tab, set the **Operating point** to periodic.
- 4 Leave the **System period** parameter as the default value of auto.
- 5 Next, set the **Frequency range** parameter to [100 50e3]. This parameter defines the lowest and highest perturbation frequencies between which the sweep is performed.
- 6 Leave the **Amplitude** parameter as the default value of 1e-3. This parameter sets the amplitude of the perturbation signal.

- 7 Choose the **Perturbation** and the **Response** blocks that should be active during this analysis. These should be the  $m'$  and  $vo'$  blocks, respectively, from Fig. 9.
- 8 Leave all the values from the **Options** and **Steady-State Options** tabs as the defaults.
- 9 Click **Accept** to accept the changes and **Start analysis** to begin the Impulse Response Analysis.

**Note:**

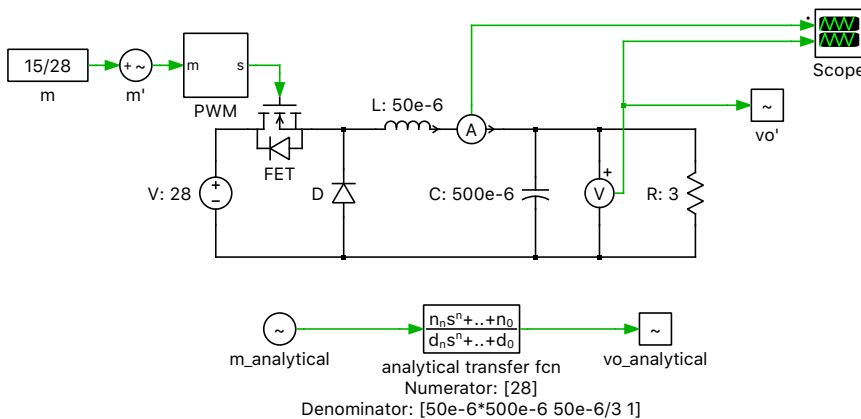
- The PLECS Analysis Tools only deliver meaningful results with switched models for frequencies up to half the switching frequency. Also, the lowest perturbation frequency will mainly influence the simulation time of the tool.
- The **Amplitude** parameter has to be small so that the system behaves in a linear manner at a certain operating point. As a general rule of thumb, the amplitude of the perturbed input should be about 1% of the input signal. If this perturbation is very small it could also make sense to lower the **Relative tolerance** value, found in the **Solver** tab of the **Simulation Parameters** dialog, accessed from the **Simulation + Simulation parameters...** menu.



At this stage, your model and simulation results should be the same as the reference analysis\_tools\_openloop\_2.plecs.

## Comparison to the analytical transfer function


There is an additional analytical transfer function block in the file analysis\_tools\_openloop\_2.plecs, as shown in Fig. 10. This is the analytical transfer function of the buck converter.



**Figure 10: Schematic of the buck converter with small signal blocks and analytical transfer function**



**Your Task:** Compare this analytical transfer function to the transfer function generated by the PLECS Analysis Tools.

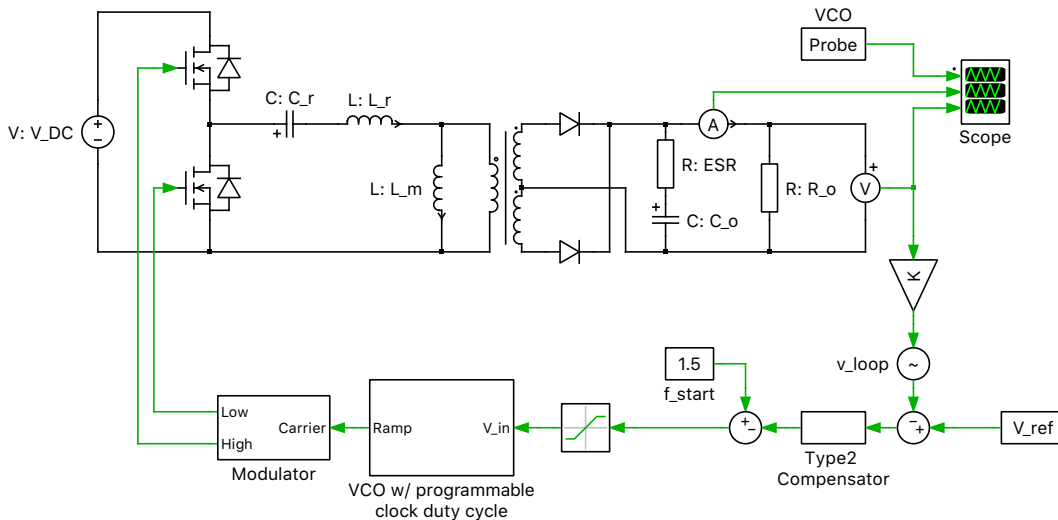
- 1 Choose the pre-configured Impulse Response Analysis, and under the **Perturbation** and the **Response** parameters, choose  $m'$  and  $vo'$  blocks, respectively, from Fig. 10. Then start the analysis. Save the current trace by clicking on the  icon in the plot window.
- 2 Next, choose the  $m_{\text{analytical}}$  and  $vo_{\text{analytical}}$  blocks in the model, that are also shown in Fig. 10, and repeat the analysis.



At this stage, you can observe that the results of the analytical transfer function and the simulation closely match.

## Loop Gain Analysis

In this section, let's calculate the open-loop gain of the half-bridge LLC converter shown in Fig. 11.



**Figure 11: Schematic of the half-bridge LLC converter with loop gain meter**



**Your Task:** Locate and open the PLECS model analysis\_tools\_loopgain\_start.plecs from the reference files.



From Tab. 1, what analysis tool that should be used to perform the loop-gain analysis of this model?



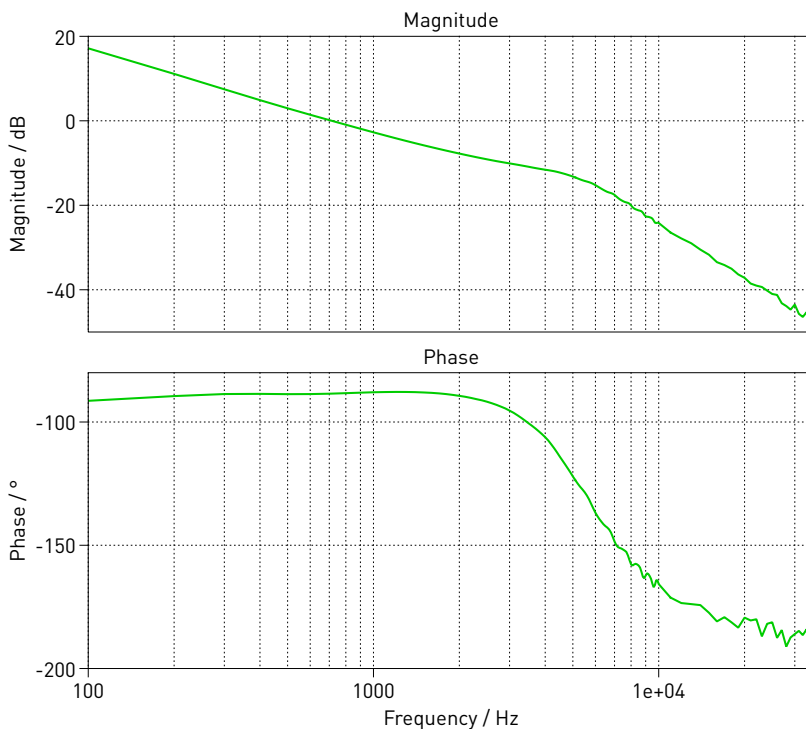
Since the controller generates a variable-frequency PWM carrier, the system period is not known prior to the analysis. In such cases the Steady-State Analysis will fail to converge, and therefore the AC Sweep tool cannot be used. The Multitone Analysis tool should be used instead.

In a Multitone Analysis, the simulation is divided into two phases. It is assumed that the system reaches its steady state in the first phase. In the second phase, the response of the system is recorded for the Fourier analysis.



### Your Task:

- 1 Add a Loop Gain Meter component to the model `analysis_tools_loopgain_start.plecs` and perform a Multitone Analysis.
- 2 First, run the model, and open the scope to observe the time when the transients of the electrical system settle out. You will notice that the system reaches steady state at around  $10e-3$  seconds.
- 3 Then from the **Setup** tab of the Multitone Analysis parameter window, set the **Init. simulation period** parameter to  $10e-3$ . This parameter corresponds to the first phase of the simulation in the description above. It is assumed that during this period, the system reaches its steady state.
- 4 Then determine the rest of the parameters as defined in Section 3.2. Note that when entering the **Frequency range** parameter, the lowest perturbation frequency you set will heavily influence the simulation time of the tool.



**Figure 12: Bode plot from the Multitone Analysis of the Half-bridge LLC converter**



At this stage, your model should be the same as the reference model, `analysis_tools_loopgain_1.plecs`. The simulation result should be as the one shown in Fig. 12.

## References

- [1] PLECS User Manual, URL: <https://www.plexim.com/download/documentation>.

## Revision History:

Tutorial Version 1.0	First release
Tutorial Version 1.1	Add information about newton iteration

## 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
	<a href="http://www.plexim.com">http://www.plexim.com</a>	Web

### *PLECS Tutorial*

© 2002–2023 by Plexim GmbH

The software PLECS described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from Plexim GmbH.

PLECS is a registered trademark of Plexim GmbH. MATLAB, Simulink and Simulink Coder are registered trademarks of The MathWorks, Inc. Other product or brand names are trademarks or registered trademarks of their respective holders.