Power Suite Filter Coefficient Tool user guide

Power Suite Filter Coefficient Tool version 1.4.0

About this document

Scope and purpose

The Power Suite Filter Coefficient Tool, (Filter Coefficient Tool/Tool), is a tool included with the ModusToolbox™ software. The Tool avails multiple capabilities among which:

- calculate device discrete regulator coefficients and generate data structures to configure an HW Filter driver
- automatically create code to support software implementation using floating-point and hardware implementation using fixed-point
- enhance the discrete regulator design flow via modeling of the resulting open-loop response using theoretical, modeled or measured plant data; there is also the possibility to add plant data, either by importing measurement data or by theoretical mathematical approach to describe the plant transfer function.

Intended audience

This document helps power and control developers to determine the filter coefficient for filter algorithms with the help of the Filter Coefficient Tool as part of creating a ModusToolbox™ application.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
Italics	Denotes file names and paths.
`Courier New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > Open	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- ADC – analog-to-digital conversion
- AWMAX – anti-windup max limit
- AWMIN – anti-windup min limit
- Application – One or more projects related to each other.
- Configurator – A GUI-based tool used to configure a resource.
- HRPWM – high resolution pulse width modulation
- MCU – microcontroller unit
- PID – proportional integration differentiation

About this document

- PWM – pulse width modulation
- TF – transfer function

Reference documents

Refer to the following documents for more information as needed:

- ModusToolbox™ tools package user guide
- Eclipse IDE for ModusToolbox™ user guide
- VS Code for ModusToolbox™ user guide
- Infineon Device Configurator user guide
- LCS Manager CLI user guide

Table of contents

1 Overview 5

1.1 Supported middleware 5

2 Launch the Filter Coefficient Tool 6

2.1 From the Device Configurator 6
2.2 make command 6
2.3 VS Code and Eclipse 6
2.4 Stand-alone (GUI) 7
2.5 Stand-alone (CLI) 7

3 Quick start 8

4 Code generation 9

5 GUI description 10

5.1 Menus 10
5.2 Toolbar 10
5.3 Set tabs 11
5.4 GUI structure 11
5.5 Notice List 11
5.6 Status bar 11
5.7 Logging 11

6 Control Panel (Compensator tab) 12

6.1 Lock/unlock Mother set 12
6.2 General 14
6.3 Input method 15
6.4 Parameters to fixed-point regulator 17
6.5 CX scaling 17
6.6 Anti-windup parameters (PSC3P6 or PSC3M6 MCU) 17
6.7 PID with Poles & Zeros input flow 17

7 Control Panel (Plant tab) 22

7.1 General (Plant tab) 22
7.2 Poles & zeros (Plant tab) 22
7.3 Import from file 23
7.4 Enable loaded data preprocessing 24
7.5 Enable Transfer function identification 24
7.6 Import open loop data from file flow 25

8 Regulator Transfer Function pane (Compensator tab) 34

8.1 Regulator TFs 34
8.2 Float-point numbers 35
8.3 Fixed-point 24-bit for HW 36

9 Plant Transfer Function pane 37

9.1 Plant TFs 37

Table of contents

9.2 Open Loop TFs 37

10 Coefficients generation flow 38

10.1 STEP 1 38
10.2 STEP 2 38
10.3 STEP 3 38
10.4 STEP 4 38
10.5 STEP 5 39
10.6 STEP 6 (Start of coefficient manipulation to fit into HW3P3Z design) 39
10.7 STEP 7 40
10.8 STEP 8 40

11 Plots (graphs) 41

11.1 Toolbar 42
11.2 Margins bar 42
11.3 Legend 42

12 Flow for compensator fast design via bode plot manipulations 44

12.1 Step 1 (Compensator gain adjustment) 44
12.2 Step 2 (Compensator poles adjustment) 45
12.3 Step 3 (Compensator zeros adjustment) 46

13 Version changes 47

1 Overview

The Filter Coefficient Tool allows you to:

- automatically create code for a variety of use cases that leverage the **3P3Z** filter after a Direct Form I implementation
- define the general parameters
- add poles and zeros defined in the frequency domain in the hertz unit
- calculate the coefficient values
- display the calculated **Regulator** coefficients of the **Discrete Transfer Function** in the **Float-point** and **Fixed-point** formats.



1.1 Supported middleware

The ModusToolbox™ Device Configurator uses the HW filter personality to launch the Filter Coefficient Tool GUI.

Name	Version	Link
mtb-dsl-psc3m8	1.2.0	GitHub - Infineon/mtb-dsl-psc3m8
mtb-dsl-psc3m6	1.0.0	GitHub - Infineon/mtb-dsl-psc3m6

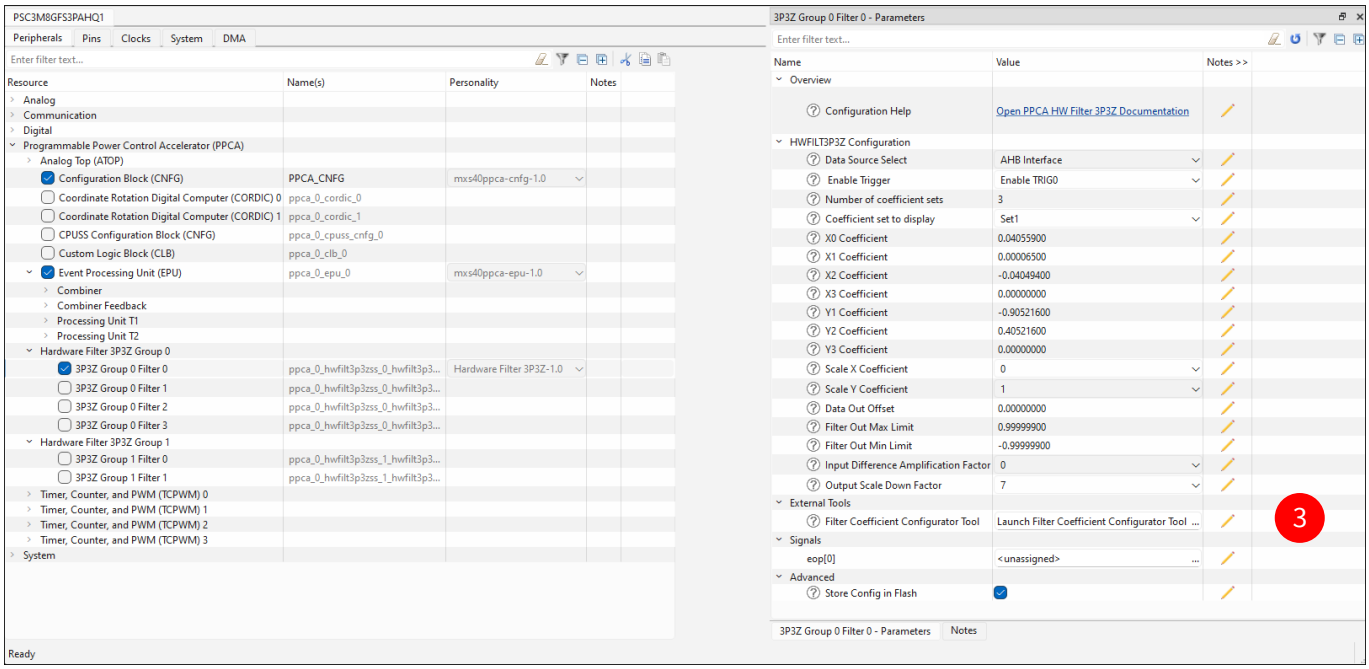
2 Launch the Filter Coefficient Tool

There are several ways to launch the Filter Coefficient Tool and those ways depend on the application and how you use the various tools.

2.1 From the Device Configurator

The Device Configurator displays information based on the *design.modus* file. Information about the device and the application is passed to the Filter Coefficient Tool. Save changes in the Filter Coefficient Tool to update the *design.modus* and *design.mtbcoef* files and generate code.

Launch the Device Configurator. (For details, refer to the Infineon Device Configurator user guide.)



1. On the **Peripherals** tab, open the **Programmable Power Control (PPCA)** resource.
2. Select **Hardware Filter** to configure the hardware-level parameters and peripherals.
3. On the **Parameters** pane, click the **Launch Filter Coefficient Tool** button.

2.2 make command

As described in the ModusToolbox™ tools package user guide " ModusToolbox™ build system" chapter, you can run numerous `make` commands in the application directory, such as launching the Filter Coefficient Tool. Create a ModusToolbox™ application, navigate to the application directory, and type the following command in the appropriate bash terminal window:

```
make filter-coef-configurator
```

This command opens the Filter Coefficient Tool GUI for the specific application in which you are working.

2.3 VS Code and Eclipse

VS Code and Eclipse have tools to launch the Filter Coefficient Tool from within an open application. Refer to the applicable user guide for more details:

- VS Code for ModusToolbox™ user guide

Launch the Filter Coefficient Tool

- Eclipse IDE for ModusToolbox™ user guide

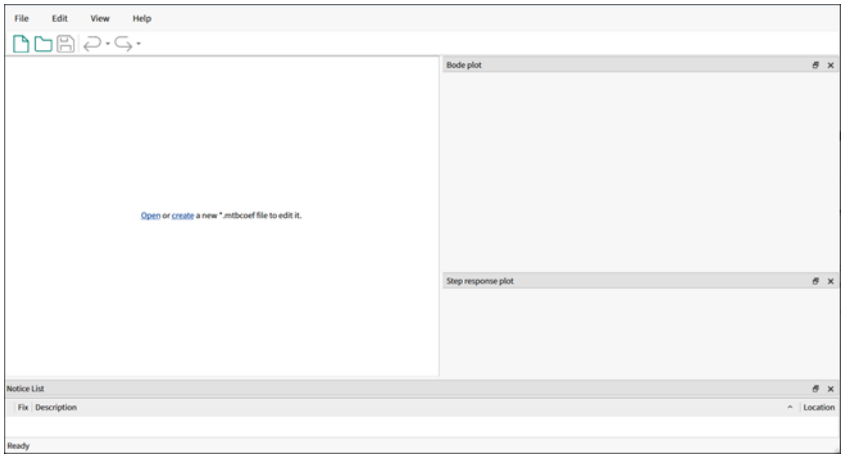
2.4 Stand-alone (GUI)

You can launch the Filter Coefficient Tool by running its executable as appropriate for your operating system (for example, double-click it or select it using the Windows **Start** menu). By default, the Tool is installed here:

C:\Users\<username>\ModusToolbox\packs\ModusToolbox-Power-Suite\tools\filter-coef-configurator

<install_dir>/ModusToolbox/packs/ModusToolbox-Power-Suite/tools/filter-coef-configurator

In stand-alone mode, the Filter Coefficient Tool opens blank. Open an existing or create a new **.mtbcoef* file and save the results.



See Menus for more information.

2.5 Stand-alone (CLI)

You can run the Filter Coefficient Tool executable from the command line. For details about the command-line options, run the filter-coefficient-configurator executable using the `-h` option.
Options:

```
-?, -h, --help               Displays help on commandline options.
  --help-all                 Displays help including Qt specific options.
  -v, --version              Displays version information.
  -d, --design <design_file> Opens or creates the *.modus
                             file provided.
  -r, --resource <resource>  Hardware resource to configure.
  -c, --config <config_file> Path to the configuration file.
```

3 Quick start

This section provides a simple workflow for how to use the Filter Coefficient Tool.

1. Launch the Filter Coefficient Tool:
   a. You can launch it from the ModusToolbox™ Eclipse IDE. The Tool will open an existing project with an empty tab ready for user inputs.
   b. You can manually launch the Filter Coefficient Tool application in stand-alone mode.
2. Configure the input parameters.

   Note: After every change, the Tool updates the result coefficients in real time.

3. Click **Save** to save the configuration and generate or regenerate the *.h* and *.c* files (respectively to item 1. a. and 1. b.):
   a. The calculated values will be transferred to the Device Configurator and it will generate the *.h*, and *.c* files.
   b. The Tool will generate its own *.h*, and *.c* files.

4 Code generation

The Filter Coefficient Tool relevant firmware will be generated depending on whether the **.modus* file is passed or not passed to the Tool:

*Note: The *.modus file is passed when the Filter Coefficient Tool is opened from the ModusToolbox™ Quick Panel or Device Configurator.*

a. If the **.modus* file is passed, code is generated to the *cycfg_peripherals.h* and *cycfg_peripherals.c* files in the *GeneratedSource* folder next to the **.modus* file;

```
const cy_stc_ppca_hwfilt3p3z_config_t ppca_0_hwfilt3p3zss_0_hwfilt3p3z_1_Set_3_config =
{
    .srcSel = CY_AHB_IF,
    .enTrig0 = true,
    .enTrig1 = false,
    .cx0 = 0x00771422,
    .cx1 = 0xFF959A42,
    .cx2 = 0xFF8941E2,
    .cx3 = 0x006ABBC2,
    .cy1 = 0x00565332,
    .cy2 = 0x002D41B7,
    .cy3 = 0xFFFBBF0E,
    .dataOutOffset = 0x00000000,
    .limMax = 0x007FFFF7,
    .limMin = 0xFF800009,
    .scaleCX = 0,
    .scaleCY = 0,
    .gIn = 0,
    .gOut = 1,
};

#if defined (COMPONENT_MTB_HAL) && (MTB_HAL_DRIVER_AVAILABLE_NPNZ)
const mtb_hal_npnz_configurator_t ppca_0_hwfilt3p3zss_0_hwfilt3p3z_1_hal_config =
{
    .base = ppca_0_hwfilt3p3zss_0_hwfilt3p3z_1_HW,
};
#endif /* defined (COMPONENT_MTB_HAL) && (MTB_HAL_DRIVER_AVAILABLE_NPNZ) */
```

b. If the **.modus* file is not passed, code is generated to the *cycfg_filtercoef.h* and *cycfg_filtercoef.c* files in the *GeneratedSource* folder next to the **.mtbcoef* file.

```
#include "cycfg_filtercoef.h"

cy_stc_ppca_hwfilt3p3z_config_t Set1_config =
{
    .cx0 = 0x00771422,
    .cx1 = 0xFF959A42,
    .cx2 = 0xFF8941E2,
    .cx3 = 0x006ABBC2,
    .cy1 = 0x00565332,
    .cy2 = 0x002D41B7,
    .cy3 = 0xFFFBBF0E,
    .dataOutOffset = 0x00000000,
    .limMax = 0x007FFFF7,
    .limMin = 0xFF800009,
    .scaleCX = 0,
    .scaleCY = 0,
    .gIn = 0,
    .gOut = 1,
};

cy_stc_filter_float_config_t Set1_float_config =
{
    .order = 3,
    .b = {
        0.9303018889,
        -0.8312301187,
        -0.9276768175,
        0.8338551901
    },
    .a = {
        1,
        -0.6744137043,
        -0.3535682124,
        0.03323206039
    },
};
cy_stc_ppca_hwfilt3p3z_config_t Set_2_config =
```

5 GUI description

The Filter Coefficient Tool GUI contains menus, toolbar, tabs for editable sequence of sets to configure the hardware filter settings, plots, and a Notice List to provide indications.

5.1 Menus

5.1.1 File

- **New** – Creates a new file with new configuration.
- **Open…** – Opens a specified configuration file (either **.modus* or **.mtbcoef*). The current file with the same extension, if any, will be closed. However, both **.modus* and **.mtbcoef* files can be opened simultaneously. Opens the folder with the configuration file.
- **Close** – Closes the current configuration file.
- **Save** – Saves the existing configuration. Saves the current configuration file and generates the middleware configuration C code.
- **Save As…** – Saves the existing file under a different name. Saves the current configuration in a new file and generates the middleware configuration C code.
- Recent files – Shows recent files that you can open directly.
- **Exit** – Closes the Tool.

5.1.2 View

- **Bode plot** – Shows/hides the Bode plot.
- **Step response plot** – Shows/hides the Step response plot
- **Notice List** – Shows/hides the Notice List.
- **Toolbar** – Shows/hides the toolbar.
- **Reset View** – Resets the view to the default.

5.1.3 Help

- **View Help** – Opens this document.
- **About Power Suite Filter Coefficient Tool** – Opens the **About** box for version information, with links to open https://www.infineon.com and the current session log files of the application and hardware configuration server.

5.2 Toolbar

Use the check box under the **View** menu to show or hide the toolbar:



The commands:

- **New**
- **Open**
- **Save & Generate**.

5.3 Set tabs



A set represents an independent filter configuration view. Use + and – on the right to add/remove a tab with a specified set of coefficients. To edit the **Set#** name, double-click it.

5.4 GUI structure

The two tabs **Compensator** and **Plant** display the **Control Panel** and various panes.

	Panel/panes							**Transfer functions panes/tables**					
Tab/panel/pane/table	**Control Panel**	**General**	**Poles & Zeros**	**Parameters to fixed-point regulator**	**CX scaling**	**Import from file**	**Loaded data preprocessing**	**TF identification**	**Transfer functions**	**Float-point numbers**	**Fixed-point 24-bit for HW**	**Plant TFs**	**Open Loop TFs**
Compensator	√	√	√	√	√	–	–	–	√	√	√	–	–
Plant	√	√	√	–	–	√	√	√	√	–	–	√	√
Plots (graphs)													

5.5 Notice List

The **Notice List** pane summarizes the presence of errors, warnings, tasks, and notes originating in different sections of the tool. Double-click a notice on the list to display the parameter, which causes the error or warning.

For more information, refer to the Infineon Device Configurator user guide.

5.6 Status bar

The bottom left corner of the **Status bar** displays additional information, for example, input field ranges. You can see such information by hovering on the most editable fields on the panel.

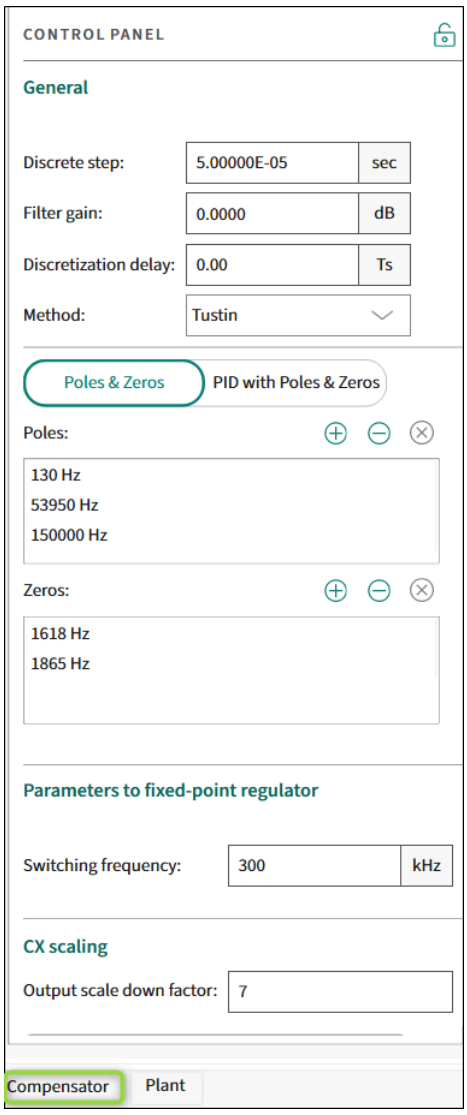


5.7 Logging

The Tool automatically generates a log for each session. The file location differs for each operating system and can be found in the **About** dialog.

6 Control Panel (Compensator tab)

The **Control Panel** consists of **General**, **Poles & zeros**, **Parameters to fixed-point regulator**, **CX scaling** panes.



6.1 Lock/unlock Mother set

Mother set is strictly coupled with the presence of an open loop response but only if it comes from **Load open loop data file**. The **Mother set** TF is used to extract **Plant** data from the loaded **Open loop** response. **Plant** data will be used to design a new compensator set to produce a new open loop.

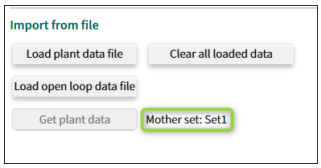
Lock

To lock the set parameters and promote the selected set to the **Mother set** status click  (top right corner). On the **Compensator** tab, only one set can be the **Mother set** at a time. The discretization delay value of the **Mother set** becomes non-editable and is forcibly propagated to all other sets.

Control Panel (Compensator tab)

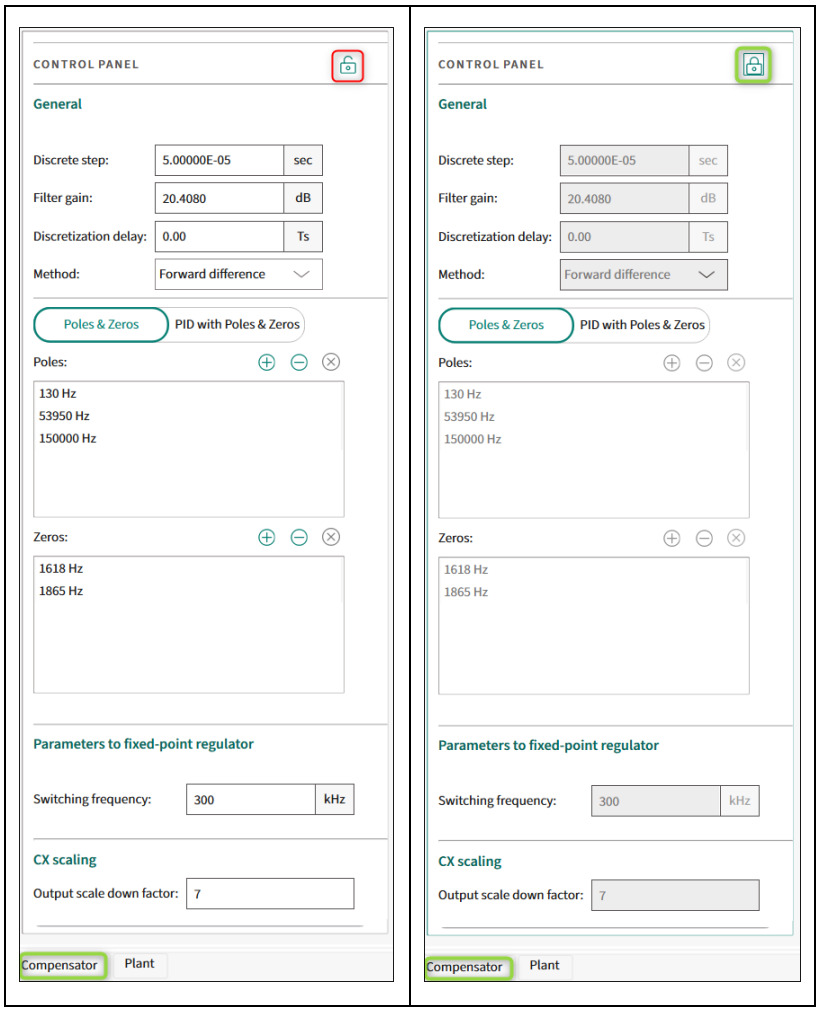
Unlock

Unlocking demotes the set to the regular status, enables the parameters editing and promotion of another set to the **Mother set** status. The name of the current **Mother set** displays on the **Plant** tab on the **Import from file** pane.



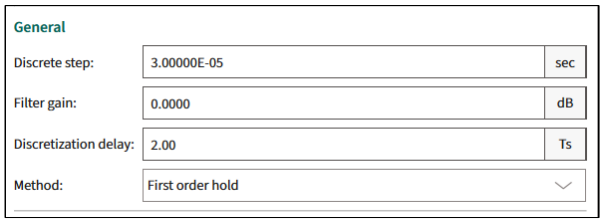
The **Plant** data extracted from the loaded **Open loop** response will be used to design a new compensator set to produce a new open loop.

Optionally, **Plant** poles and zeros can be identified from **Plant** data and used for the same design purposes (refer to **Poles & zeros (Plant tab)**).



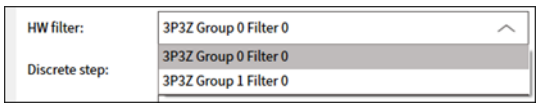
6.2 General

The **General** pane consists of the **HW filter**, **Discrete step**, **Filter gain**, **Discretization delay** parameters and **Method** select field.

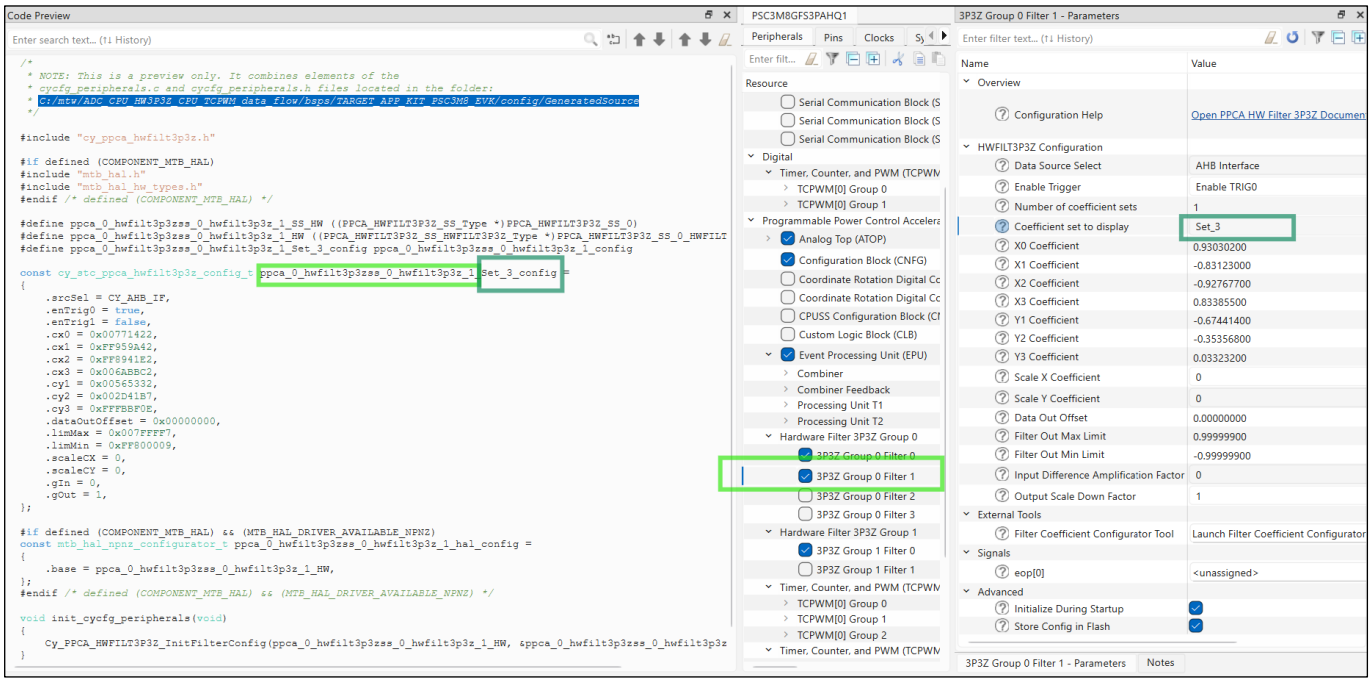


6.2.1 HW filter

Available only for a ModusToolbox™ project created for an MCU product that supports HW filters. In the Device Configurator, enable **Hardware Filters**. In the Filter Tool, select a filter from the enabled filters. The configured set will be applied to the selected filter. It is possible to reassign a set by selecting another **HW filter** in the Filter Tool.



In the Device Configurator, the set name displays on the configuration pane and used in the naming (right after HW filter group) of the data structure with the calculated control coefficients.



6.2.2 Discrete step

The **Discrete step** (sec) field represents the discretization time to translate the Continuous filter Transfer Function into its Discrete form. It is initialized with default value 3.33e-06 representing a control frequency of 300 kHz. The Discrete step is updated and then its value edited after you change the switching frequency parameter. The Discrete step value range: 1e-07 to 1 sec.

6.2.3 Filter gain

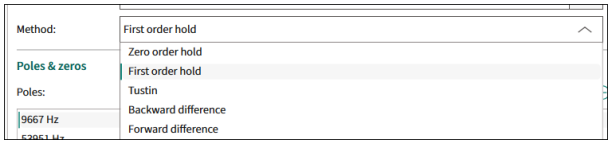
The **Filter gain** field represents the gain value of the filter, which shifts the Regulator amplitude response “up” or “down”. The default value is 1 dB. The filter gain value range: -500 to 500 dB.

6.2.4 Discretization delay

The time expressed in fraction of a PWM period to account for the inherent delay introduced by digital implementation – the delay introduced from the digital implementation from sensing to effective actuation. The delay is accounted in the phase margin determination.

6.2.5 Method

The **Method** field represents the discretization methods used to translate the filter Transfer Function to its Discrete form. The list of available options:



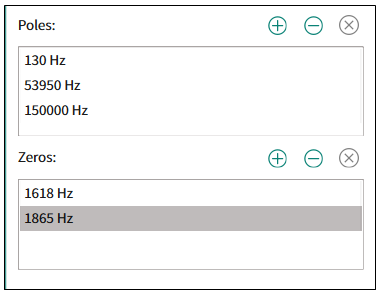
6.3 Input method

The available input modes:

- **Poles & Zeros** – Generic with up to 5 poles/zeros.
- **PID with Poles & Zeros** – Focused for **PID** filter.

*Note: Switching **Poles & Zeros** to **PID with Poles & Zeros** erases **Poles & Zeros** mode inputs and populates all parameters with default values.*

The Tool determines the Regulator coefficients for the Regulator Discrete form representation starting from poles and zeros placement in the **Continuous** domain.

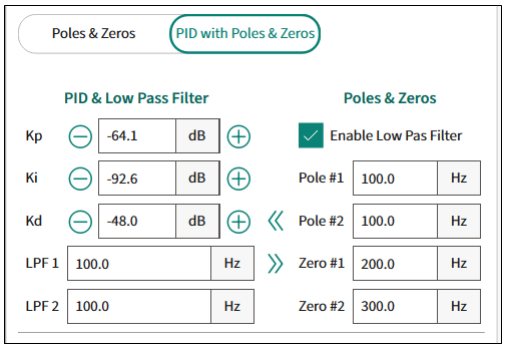


Use  and  on the right to add/remove poles and zeros. Use  to clear all poles and zeros. Double-click the value to edit it.

6.3.1 Poles & zeros input mode

Compensator poles and zeros are entered in Hertz (Hz) within the range -1×10^{24} to $+1\times10^{24}$. The real poles/zeros are entered as: single numbers for example, -100; when complex, in pairs using the j notation (for example, -100+200j and -100-200j). Enter complex poles and zeros as conjugate pairs—the tool validates this automatically and displays an error if the pairs are incomplete. The values are sorted automatically by magnitude.

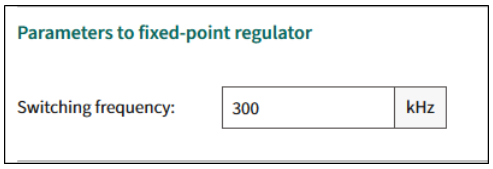
6.3.2 PID with Poles & Zeros input mode



Parameter	Description	Range	Default
Kp	Proportional gain – sets the response to the present error, speeds up correction.	-500 - 500 dB	-40.6 dB
Ki	Integral gain – adds correction based on the accumulated error; so, helps remove the steady-state error.	-500 - 500 dB	-92.6 dB
Kd	Derivative gain – responds to how quickly the error changes; so, helps counteract the error slope by enabling a faster response.	-500 - 500 dB	-1.0 dB
LPF 1/LPF 2	The same values as for Pole#1 and Pole #2. Set the two low-pass filter pole frequencies for PID compensator. These poles attenuate high-frequency components, so, help reduce noise.	0 – 1e9 Hz	100 Hz
Enable/Disable Low Pass Filter	Click **Enable Low Pass Filter**		
Filter gain	Calculated from **Ki**, for details, refer to section 6.7.2.1.		
Pole #1/Pole #2	The same values as for LPF 1 and LPF 2. Set the two low-pass filter pole frequencies for PID compensator. These poles attenuate the high-frequency components, which helps reduce noise.	0 – 1e9 Hz	100 Hz
Zero #1	The real zero of the PID controller numerator. Its frequency is calculated in Hz from PID gains and determines where the controller response changes along with the frequency.	0 – 1e9 Hz	200 Hz
Zero #2	The real zero of the PID controller numerator. Its frequency is calculated in Hz from the PID gains and determines where the controller response changes along with the frequency.	0 – 1e9 Hz	300 Hz

 – Use these arrow buttons to recalculate **Poles & Zeros** to **PID** and **PID** to **Poles & Zeros** after parameters editing. Refer to section 6.7.1 for the calculation flow and an example.

6.4 Parameters to fixed-point regulator



Parameter	Description	Range	Default
Switching frequency	The target switching frequency of power switch operation	20 - 2000 kHz	300 kHz

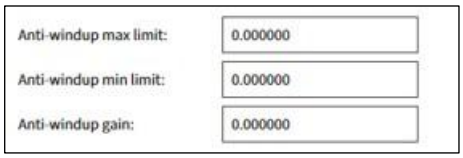
6.5 CX scaling



Note: Changing G_OUT to 1 will lower the Regulator amplitude by 6 dB.

Parameter	Description	Range	Default
Output scale down factor	The **Output scale down factor** affects an additional attenuation on the output of the filter through the G_OUT factor.	0 to 7	0

6.6 Anti-windup parameters (PSC3P6 or PSC3M6 MCU)



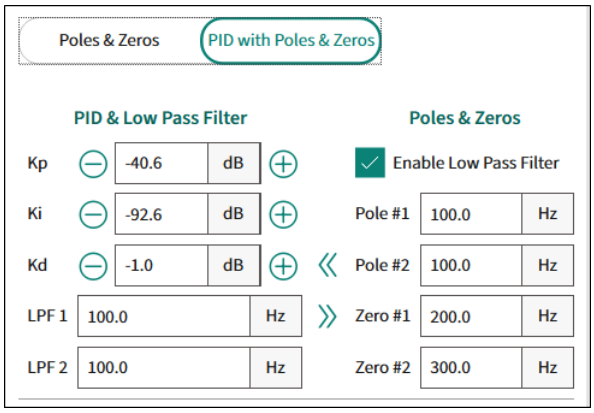
Parameter	Description	Range	Default
Anti-windup max limit	You can disable this feature by setting the Anti-windup limits registers **AWMAX** and **AWMIN** to their maximum absolute values and gain value to 0.	-0.999999 – 0.999999	0
Anti-windup min limit		-0.999999 – 0.999999	0
Anti-windup gain		-0.999999 – 0.999999	0

6.7 PID with Poles & Zeros input flow

This section describes the flow for input mode **PID with Poles & Zeros**.

6.7.1 Step 1

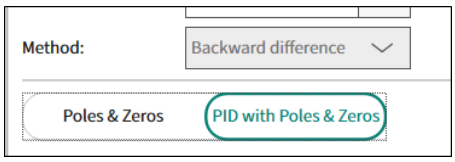
Open the Filter Coefficient Tool and click **New**. Switch the input mode to **PID with Poles & Zeros**. Create a valid **PID** representation as described below.



The parameters are populated with default values and **Low Pass Filter** is enabled.

Note: The default poles and zeros values are given for demonstration purpose.

The TF discretization method on the **General** pane will auto switch to “Backward difference” and remain non-editable.



6.7.2 Step 2

Enter/edit the **PID** parameters. Click » to recalculate the zero frequencies and filter gain. When editing poles, zeros or filter gain, click « to recalculate the **PID** parameters.

6.7.2.1 PID to poles and zeros conversion

The displayed **PID** gains values are in dB. The Tool converts them to linear digital gain:

$$K_p = 10^{K_{p,dB}/20}, K_i = 10^{K_{i,dB}/20}, K_d = 10^{K_{d,dB}/20}$$

For sampling period T_s (Discrete step), the conversion uses the following analog-equivalent gains:

$$K_{pa} = K_p, K_{ia} = \frac{K_i}{T_s}, K_{da} = K_d T_s$$

The two zeros are the roots of $K_{da}s^2 + K_{pa}s + K_{ia} = 0$ expressed as positive frequencies in Hz:

$$f_{z1} = \frac{1}{2\pi} \left| \frac{-K_{pa} + \sqrt{K_{pa}^2 - 4K_{da}K_{ia}}}{2K_{da}} \right|, f_{z2} = \frac{1}{2\pi} \left| \frac{-K_{pa} - \sqrt{K_{pa}^2 - 4K_{da}K_{ia}}}{2K_{da}} \right|$$

Control Panel (Compensator tab)

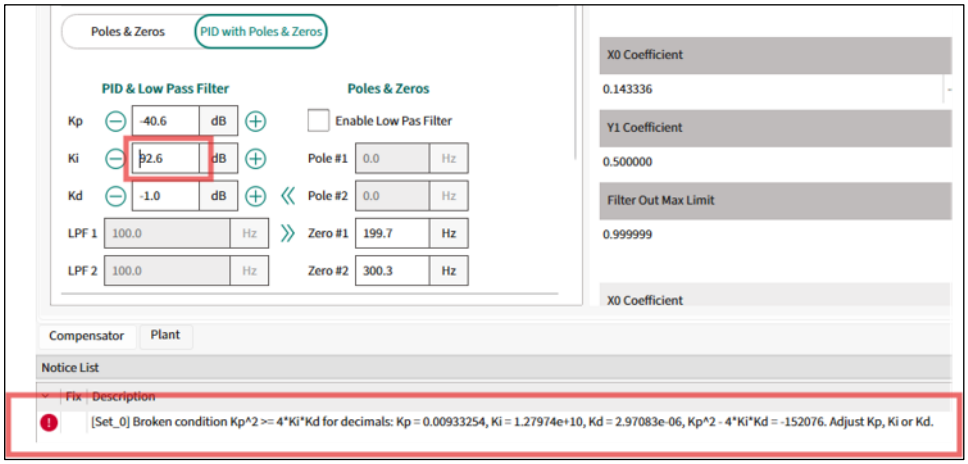
The Compensator poles are the integrator pole at the origin and, when the Low Pass Filter is enabled, the two LPF poles at **LPF 1** and **LPF 2**.

The Compensator filter gain is derived from the integrator gain:

$$f_{p0} = \frac{K_{ia}}{2\pi}, \qquad FilterGain_{dB} = 20\log_{10}(f_{p0})$$

The forward conversion requires real zeros.

If $K_{pa}^2 - 4K_{da}K_{ia} < 0$, the conversion cannot produce a real-zero representation and the Tool displays an error message with the K_p, K_i, K_d values in decimals.



6.7.2.2 Poles and zeros to PID conversion

If the zero frequencies are f_{z1} and f_{z2}, the sampling period T_s and the filter gain in dB, the Tool first converts the filter gain to the linear form:

$$f_{p0} = 10^{FilterGain_{dB}/20}$$

Calculate the linear **PID** gains:

$$K_p = f_{p0}\left(\frac{1}{f_{z1}} + \frac{1}{f_{z2}}\right)$$

$$K_i = 2\pi T_s f_{p0}$$

$$K_d = \frac{f_{p0}}{2\pi T_s f_{z1} f_{z2}}$$

Convert the gains to the dB values used in **PID** input mode:

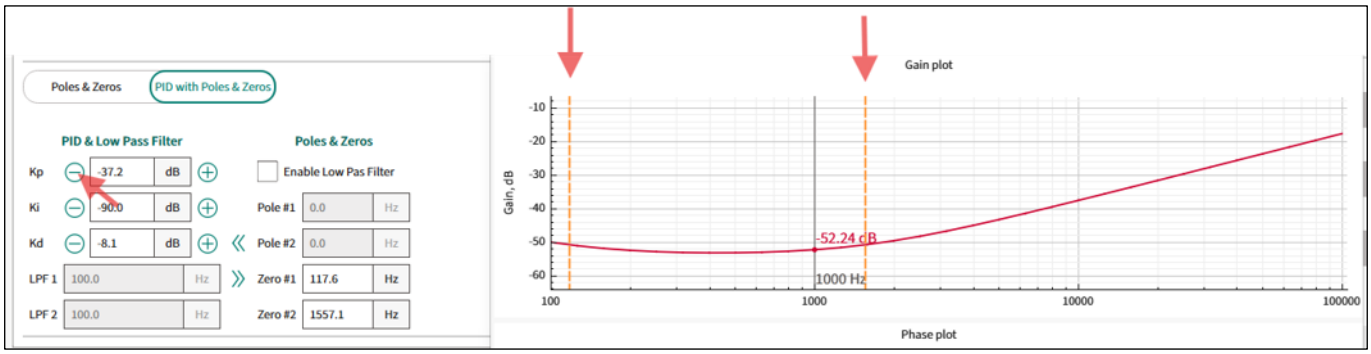
$$K_{p,dB} = 20\log_{10}(K_p), K_{i,dB} = 20\log_{10}(K_i), K_{d,dB} = 20\log_{10}(K_d)$$

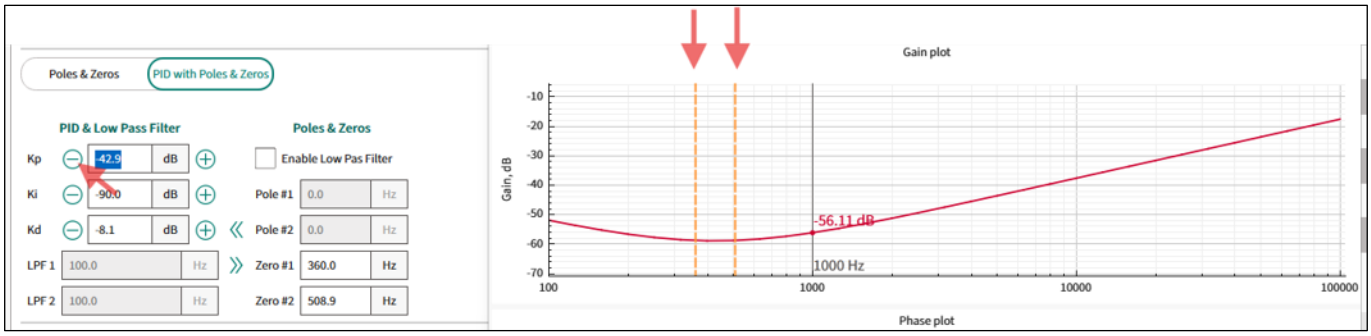
When the Low Pass filter is enabled, **LPF 1** and **LPF 2** are set from the two supplied pole frequencies. The zero frequencies, and the LPF pole frequencies when enabled, must be greater than zero.

6.7.3 Step 3

To change the Continuous **PID** parameters with a single step of 0.1 dB, left press and hold mouse button on ⊖ or ⊕ control. The Tool automatically recalculates new zeros and updates the poles and zeros along with **Bode** plot. To visualize the computed zeros and gain, click ».

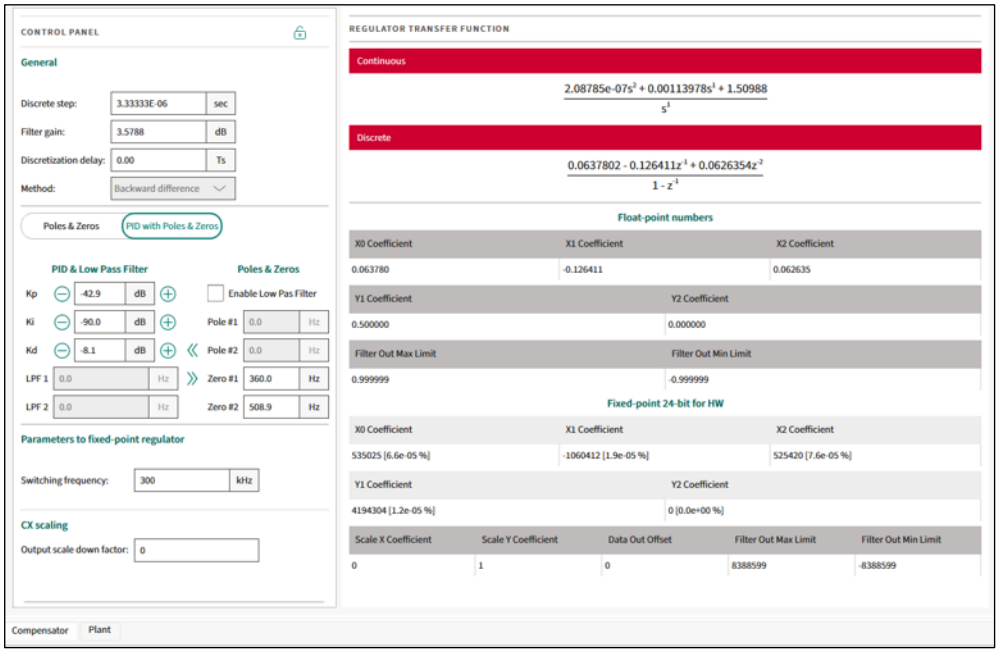
Control Panel (Compensator tab)





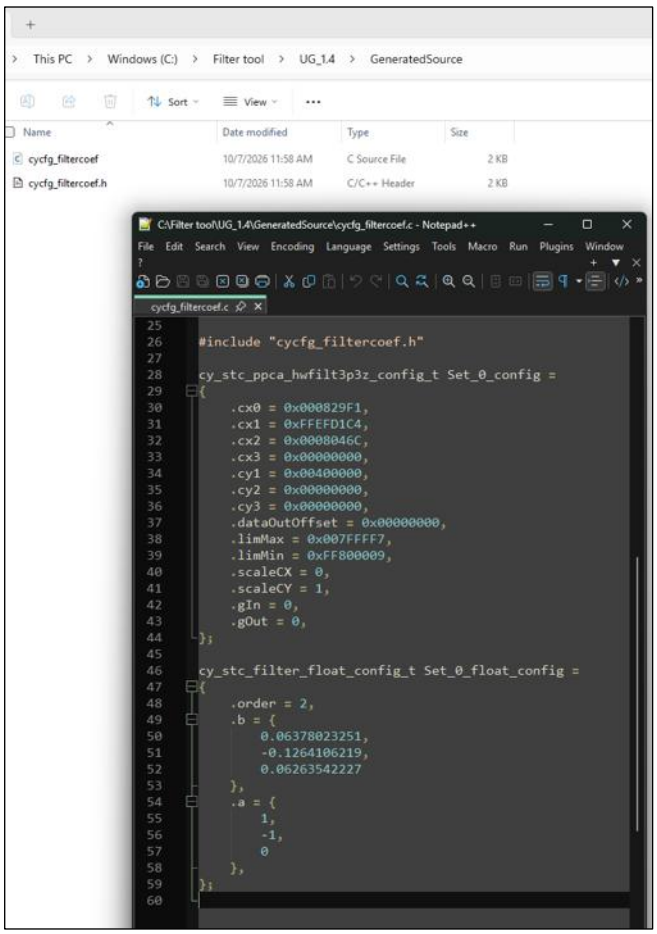
6.7.4 Step 4

The final Continuous and Discrete TF is generated. The float-point numbers table is populated with control coefficients.



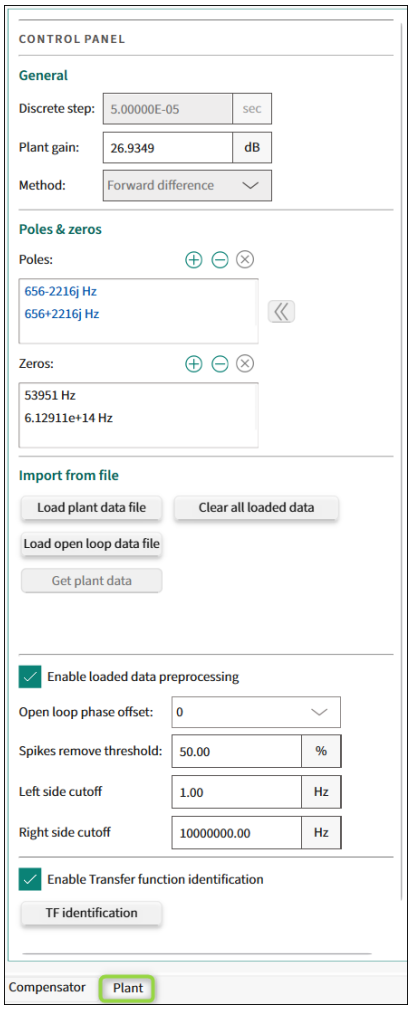
Control Panel (Compensator tab)

After saving the file in the selected folder, the Tool adds new folder *GeneratedSource* with *cycfg_filtercoef.h* and *cycfg_filtercoef.c* files. The calculated coefficients are stored in the structures to be copied and used.



7 Control Panel (Plant tab)

The **Plant** tab consists of **General**, **Poles & zeros**, **Import from file** panes.



7.1 General (Plant tab)

The **General** pane consists of the **Discrete step**, **Filter gain**, and **Method** field.

7.2 Poles & zeros (Plant tab)

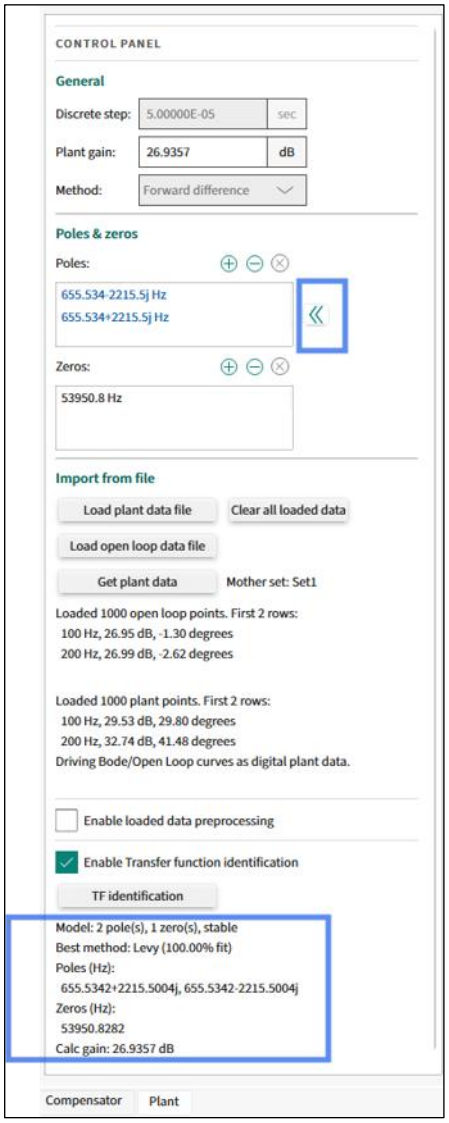
Refer to **Poles & zeros input mode**.

Plant **Poles & Zeros** define the Plant Transfer Function, which allows the Tool to:

- compute the Open Loop Transfer Function (Plant × Compensator)
- calculate the closed-loop system response (feedback system behavior)
- visualize combined Bode plots showing how compensator interacts with the Plant
- visualize step response to predict settling time, overshoot, stability margins
- design compensators that achieve desired control objectives (stability, bandwidth, phase margin).

7.3 Import from file

This section contains the **Load plant data file**, **Load open loop data file**, **Clear all loaded data**, **Get plant data** commands, which allow you to identify the Plant Transfer Function characteristics from modeled or measured frequency response data.



To populate Plant poles, zeros, and gain with the values from TF identification result, click  next to **Poles & Zeros** pane. The button is active only when TF identification result is available.

7.3.1 Load plant data file/Load open loop data file

Import Plant measurement data from a *.csv* or *.txt* file with the three columns: Frequency (Hz), Gain (dB), and Phase (degrees).

- Supported formats: *.csv*, *.txt*.
- Data points are validated. Spikes and duplicates are removed during import.
- Upon success, the first two rows of data are displayed for verification.
- The loaded data displays as scatter points on the Bode plot.
- Data size requirements: min 10, max 100000 rows.

7.3.2 Clear all loaded data

Use this to remove all currently loaded measurement data from memory and clear the visualization and information generated by TF identification. Before deletion, the confirmation dialog prompts to prevent accidental data loss.

7.3.3 Get plant data

Gets a Plant response from the loaded open-loop measurement by subtracting the locked **Mother set** Compensator gain and phase at each frequency. The resulting Plant data is stored and displayed for **Bode** analysis and Transfer-Function identification. **Mother set** Compensator must be locked and open-loop data file loaded.

7.4 Enable loaded data preprocessing

Enables preprocessing of imported open-loop data.

The detected gain and phase spikes are replaced by interpolated values. The data is restricted to the selected frequency range. The configured open-loop phase offset is also applied before the data is used for the Plant extraction and analysis.

Parameter	Description	Range	Default
Open loop phase offset	Applies the selected phase correction to all imported open-loop data points. Use 0° for no correction or -180° to shift the phase by 180 degrees.	0° or -180°	0°
Spikes remove threshold:	Sets the percentage threshold used to detect spikes in the imported gain and phase data. Points with abrupt direction-changing changes above this threshold are identified and replaced by interpolated values.	0 – 100 %	50 %
Left side cutoff	Sets the lower frequency limit for preprocessing imported data. Data points below this frequency are excluded from further analysis.	1 – 1000 Hz	100 Hz
Right side cutoff	Sets the upper frequency limit for preprocessing imported data. Data points above this frequency are excluded from further analysis.	10 kHz – 10 MHz	10 MHz

7.5 Enable Transfer function identification

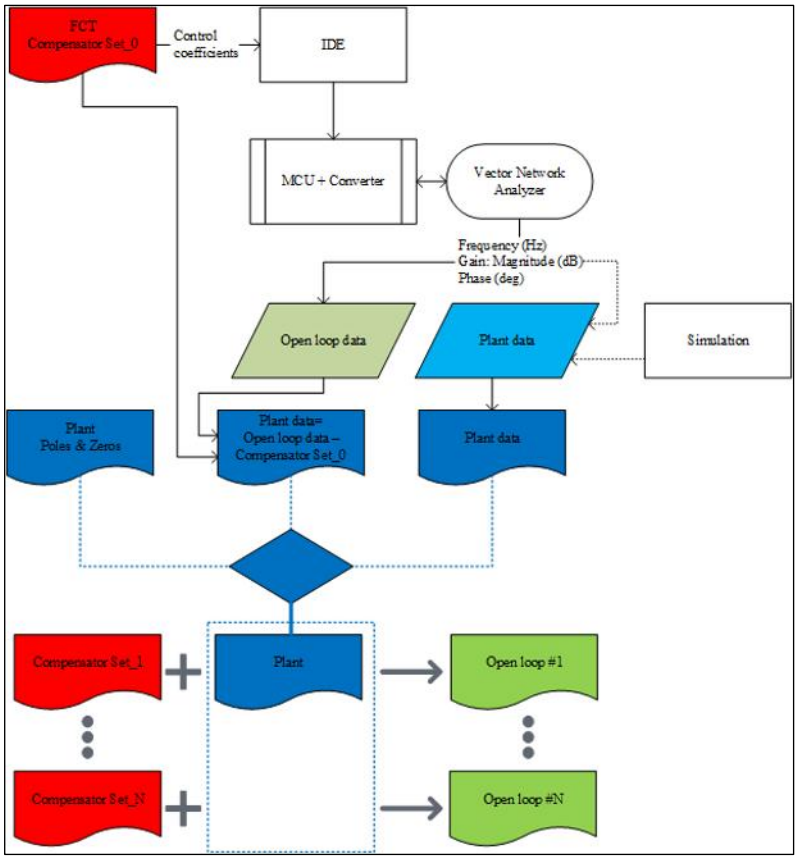
- **TF identification** – Performs the automated Transfer Function identification from the loaded frequency-response data using the system identification algorithms.

Note: Load valid measurement data before running the TF identification.

Note: TF identification does not always give high fit percentage due to internal limitations and max 5 poles and zeros.

7.6 Import open loop data from file flow

This section describes import, preprocessing, and Plant data extraction from imported Open loop data.



Plant can be populated in the following ways:

- Plant **Poles & Zeros**
- Plant data from simulation or measurement
- Plant data extracted from the measured Open loop data – the most complex, therefore described in the next steps and covers the first two points as well.

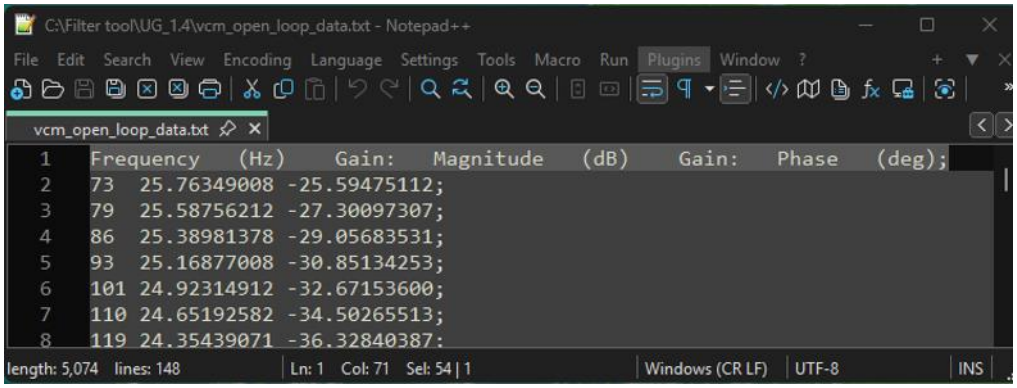
7.6.1 Step 1

Create a new file and populate the **Compensator** parameters to build valid configuration. Build a project with the calculated control coefficients and run it on the converter.



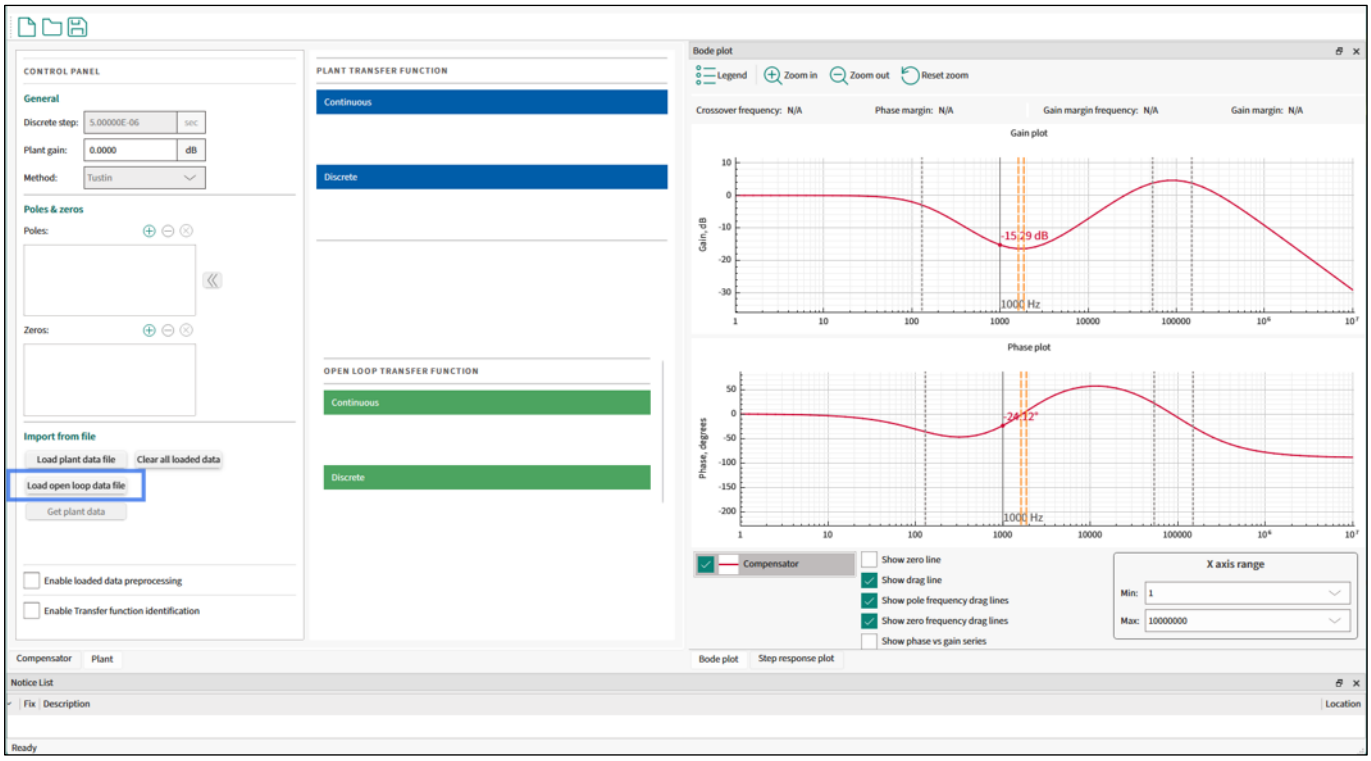
7.6.2 Step 2

Measure the Open loop response and prepare a *.csv* or *.txt* file with the data formatted in the three columns: **Frequency** in Hz, **Magnitude** in dB, **Phase** in degrees.



7.6.3 Step 3

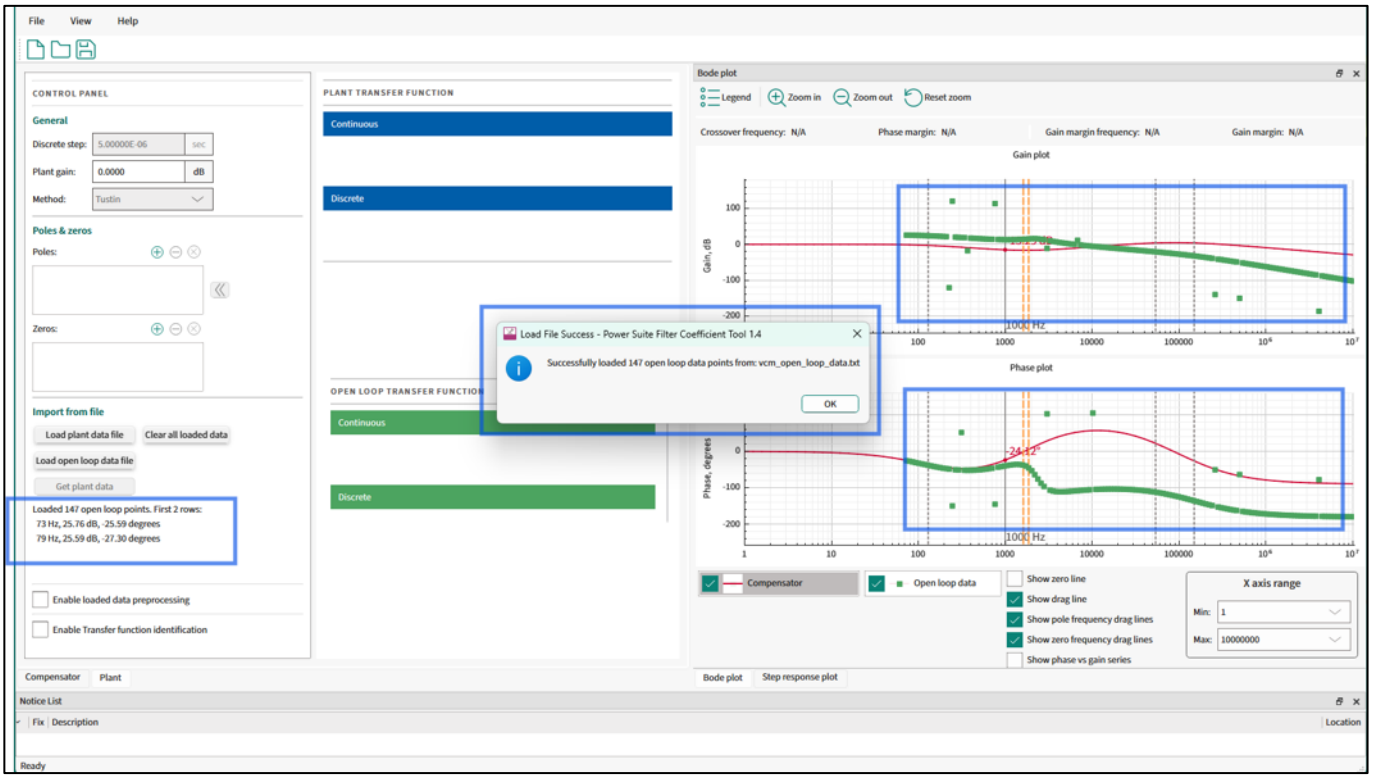
Go to the **Plant** tab and click **Load open loop data file**. Select the prepared file.



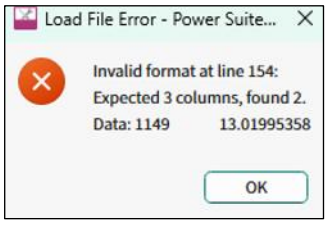
Note: The Discrete step and discretization method are non-editable. Their values are inherited from the Compensator set for consistency to build Open loop Discrete TF.

7.6.4 Step 4 (Enable loaded data preprocessing)

Ensure that the data loading is successful. The message box displays the number of unique loaded rows. The data sample of the two first rows will display on the Import from file pane. Scatter plots (green color squares) are added to **Bode** plot.



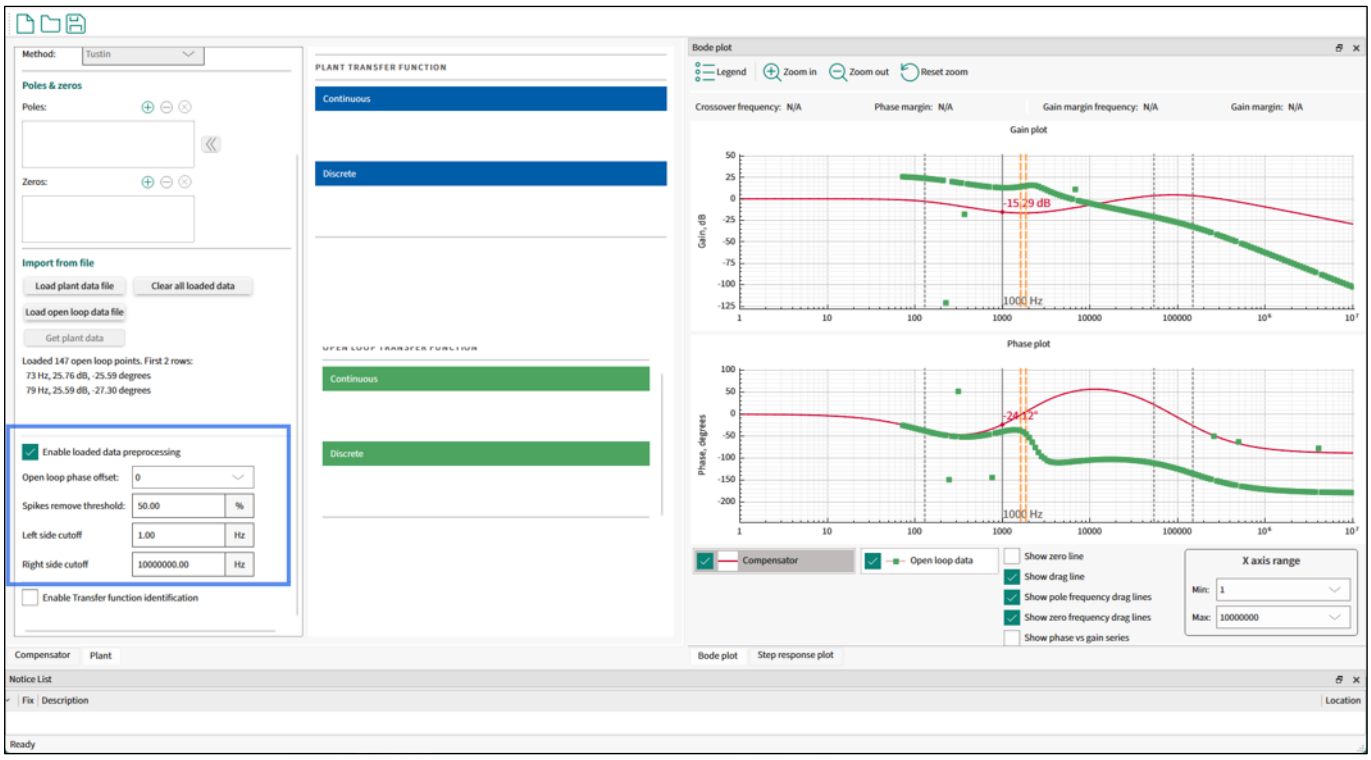
If data is not formatted in the three columns, the error message displays notifying of the corrupted line and incorrect formatted data.

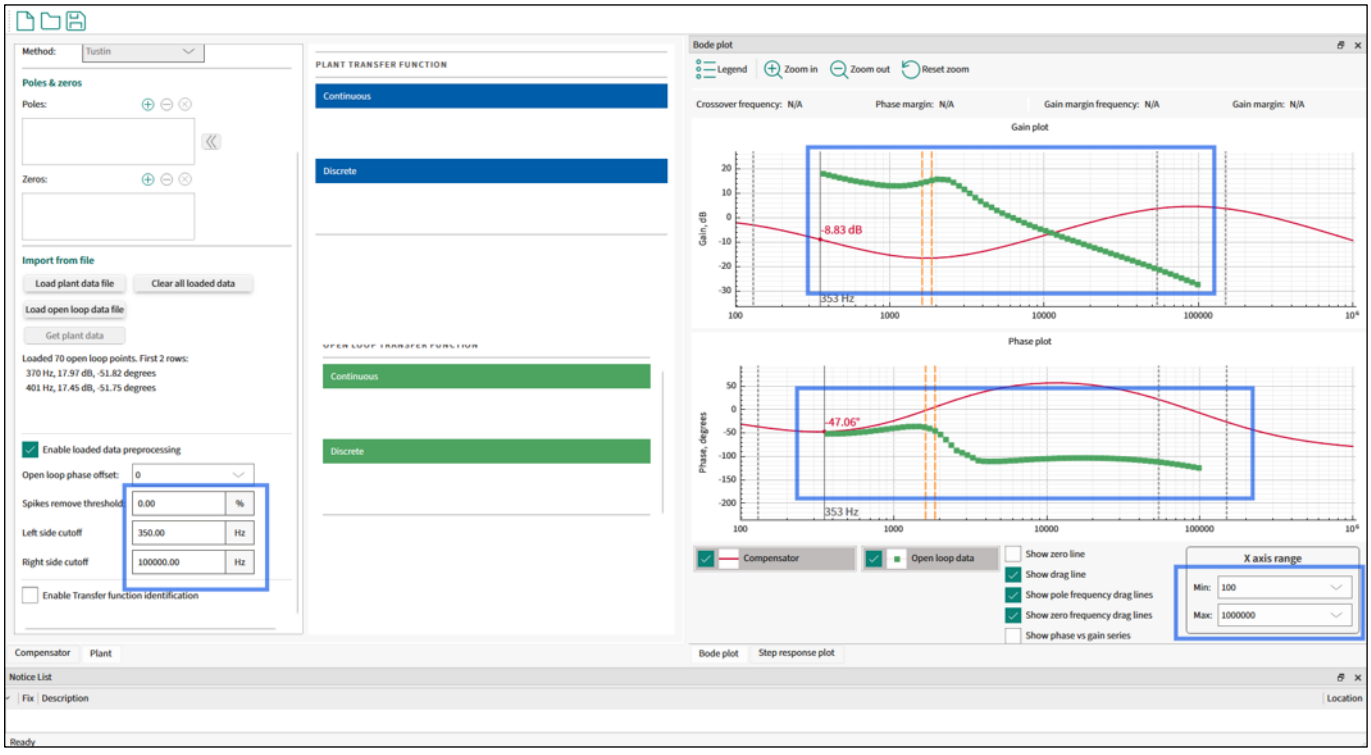


7.6.5 Step 5 (Enable loaded data preprocessing)

You can prepare and preprocess data with dedicated powerful programs before importing data to the Filter Coefficient tool.

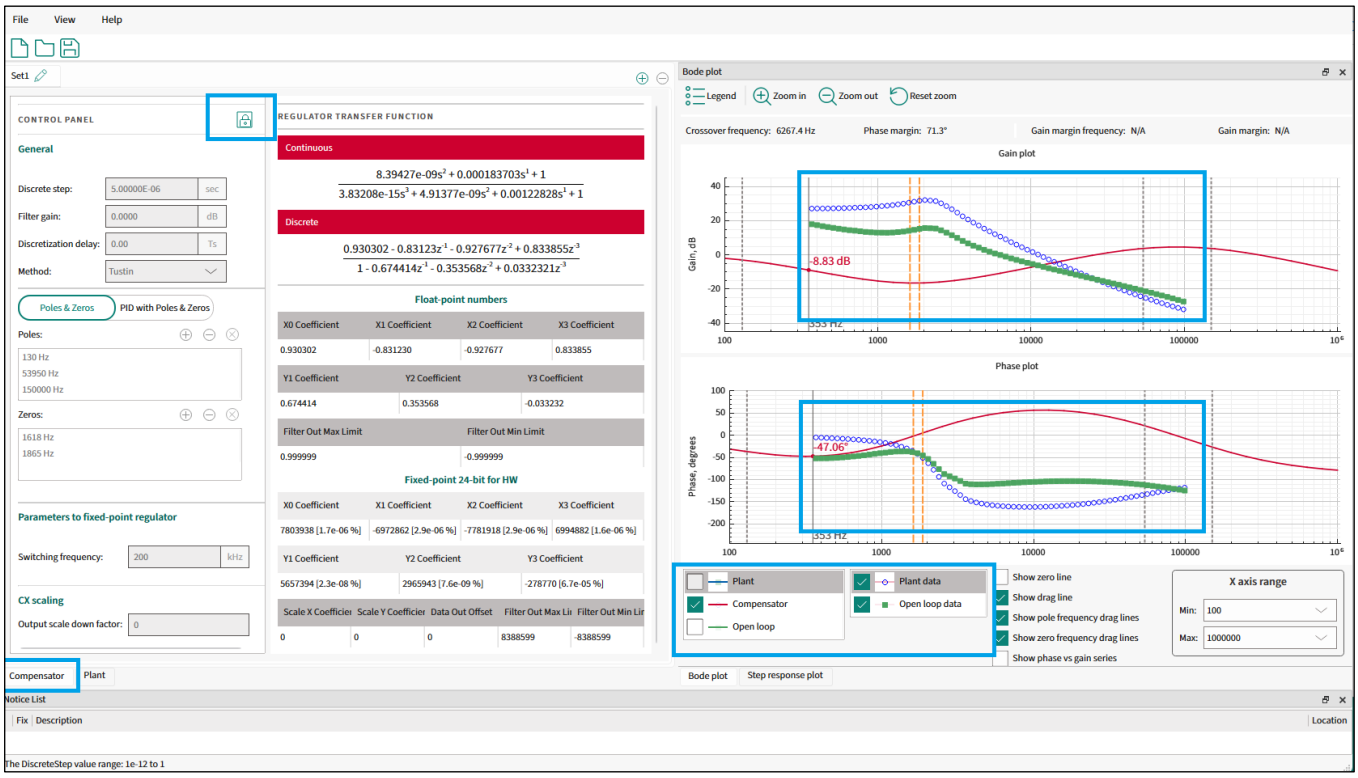
Click **Enable loaded data preprocessing**. Configure the spikes remove and cutoffs. For optimal plots visualization, select the **Bode** plot min and max **X**. Change the **Open loop phase offset** if needed.





7.6.6 Step 6 (Plan data extraction)

The **Get plant data** button is inactive. Define which set was used to receive the Open loop data. Go to the **Compensator** tab and switch to the defined set. To promote the set to the **Mother set** status, click the **Mother set** lock button.



Now, the **Control Panel** parameters are non-editable. The **Bode** plot is updated with the scatter plots of the calculated Plant data points. The extracted points (blue open circles) are at the same frequencies as the loaded Open loop. The Plant data size and range are the same as for the loaded Open Loop data.

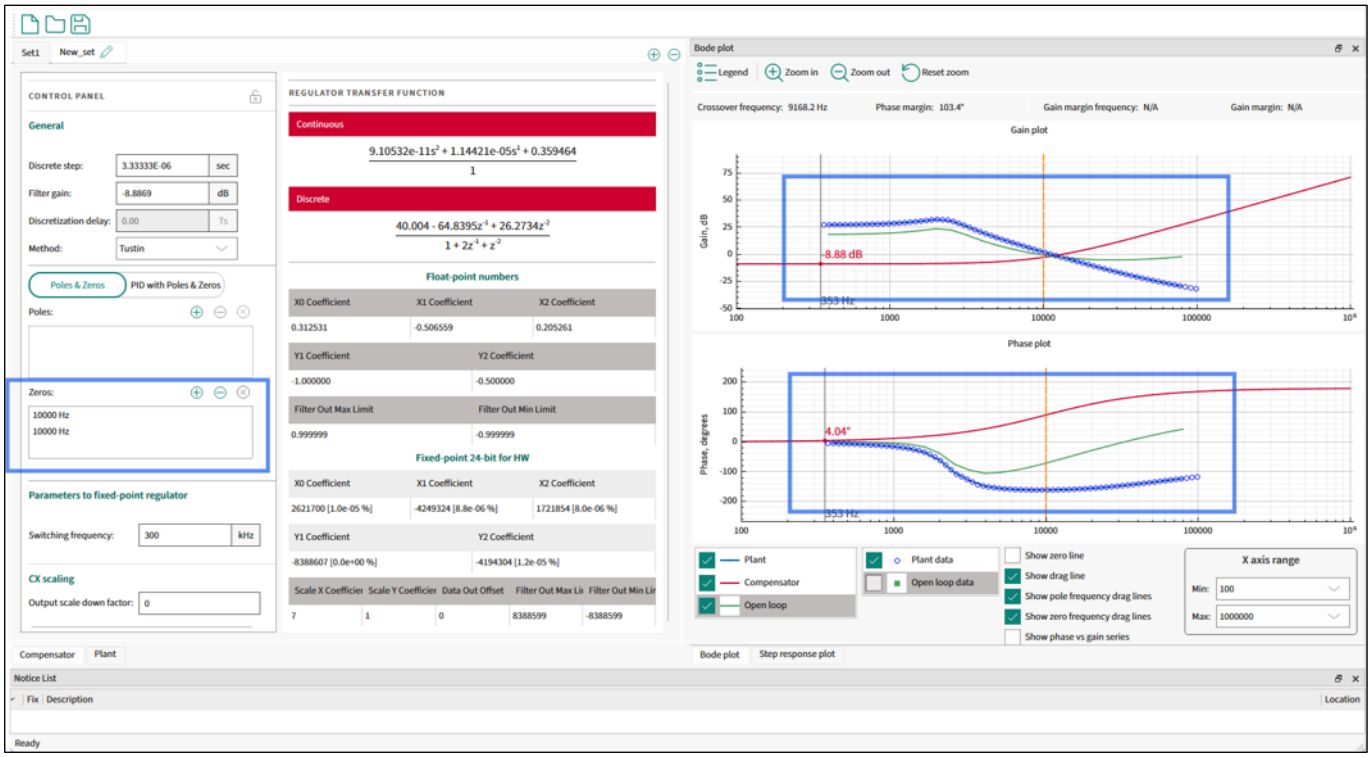
The Plant curves calculated from the Plant points:

- have higher density
- match the Compensator and calculated Open loop frequency points on the **Bode** plot
- available for a new set to be used for the analytical calculation of Open loop **Bode** plots.

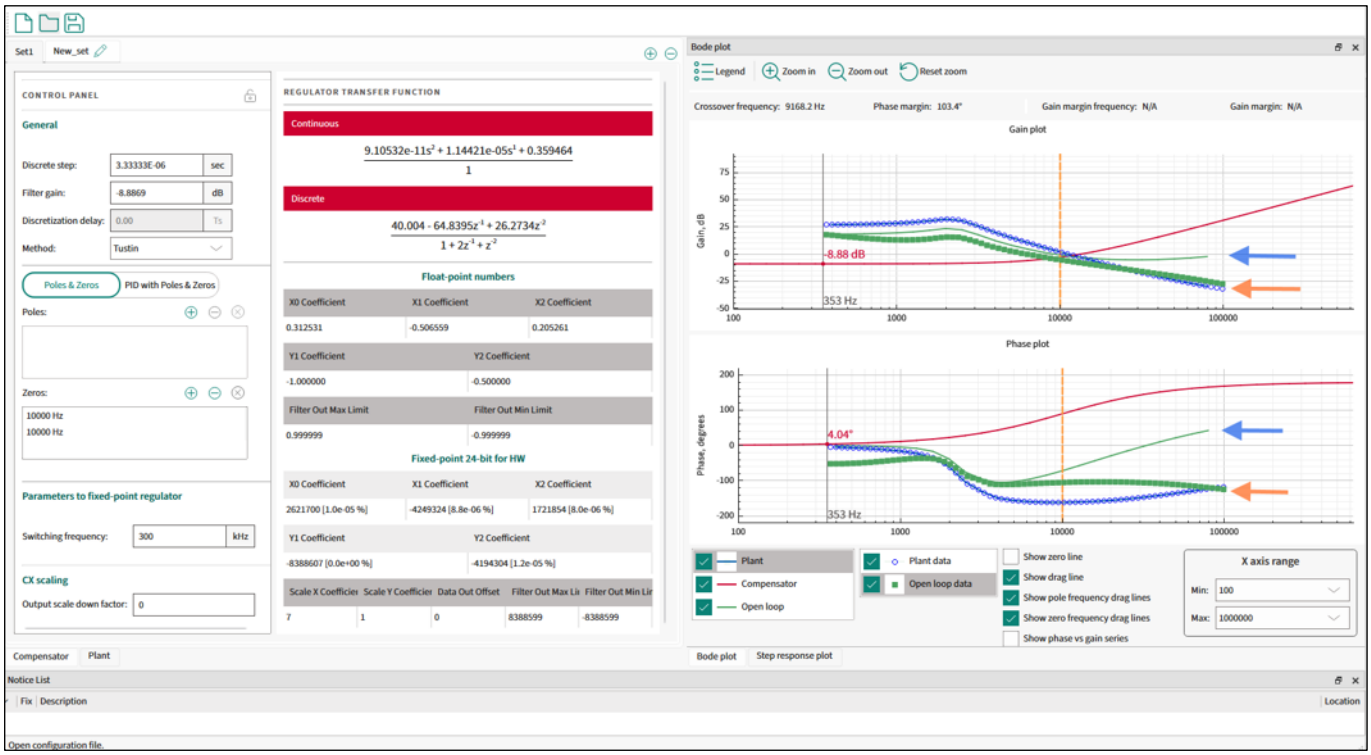
The **Bode** plot legend provides the full visualization control to display any combination of the data of three lines and two scatters.

7.6.7 Step 7

Create a new set and populate the parameters to create for example, a high pass filter. The extracted Plant will be used to calculate the Open loop for **Bode** plots.



The new Open loop response calculated with Compensator's **New_set** and the extracted Plant is available for comparison alongside the loaded Open loop data. The Tool can support up to 20 sets; thus, 19 sets can be added to exercise with the Compensator design effect on the Open loop.



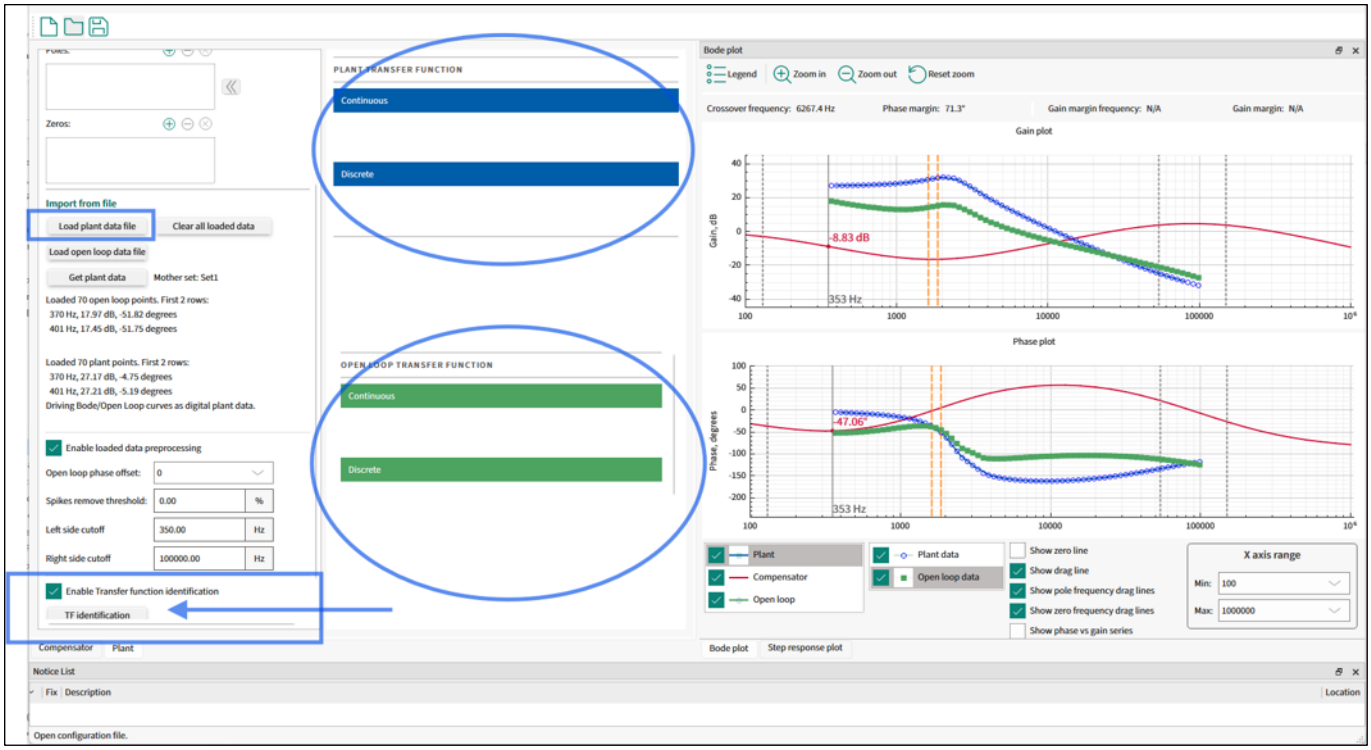
7.6.8 Step 8 (Plant TF identification)

Go to the **Plant** tab and enable the Transfer Function identification block. Then, click the **TF identification** button.

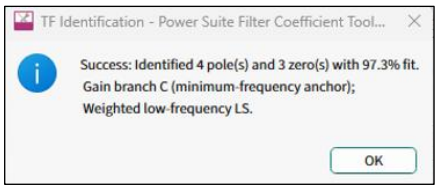
Plant data is required to perform the identification. To do this, use:

- extracted Plant data from the measured Open loop data
- Plant data from simulation or measurement loaded with the **Load plant data** button.

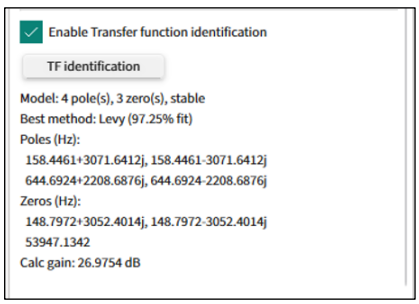
At this step, TF panes for Plant and Open loop are empty, because the Tool does not have Plant analytical representation.



The message box will inform you of the identification result:

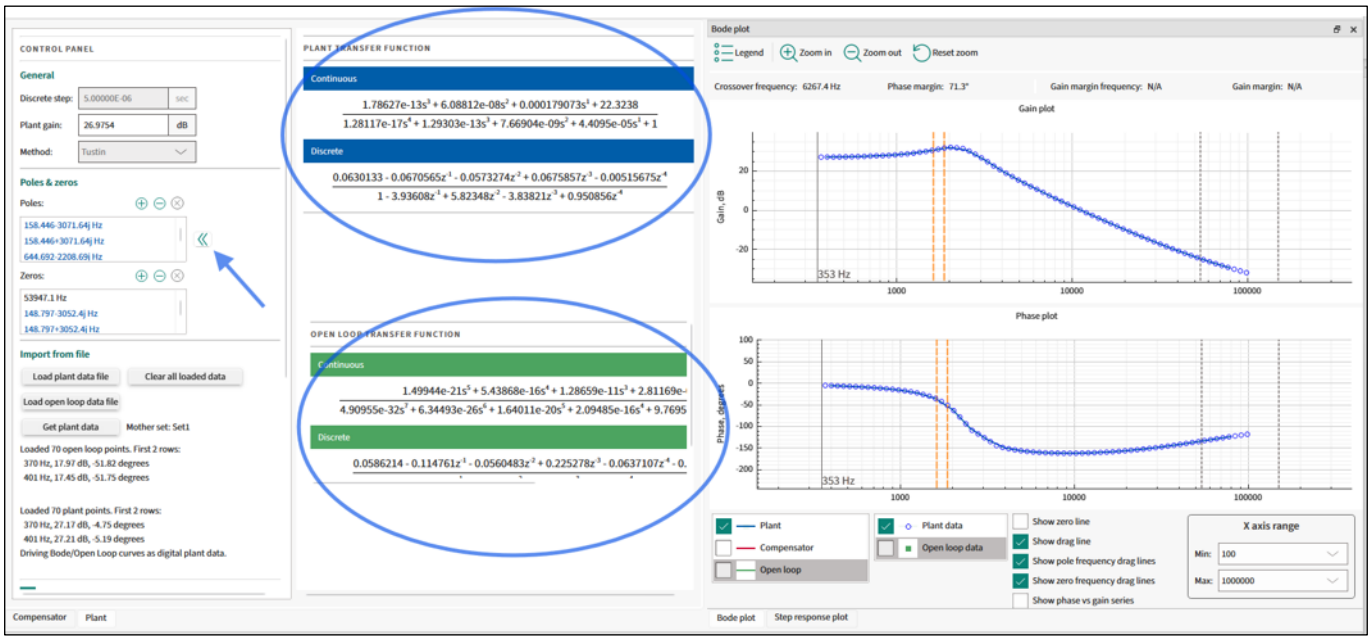


The detailed results are displayed below the **TF identification** button: the number of poles and zeros, result stability, method used for approximation, fitting percentage, poles and zeros values, and gain.

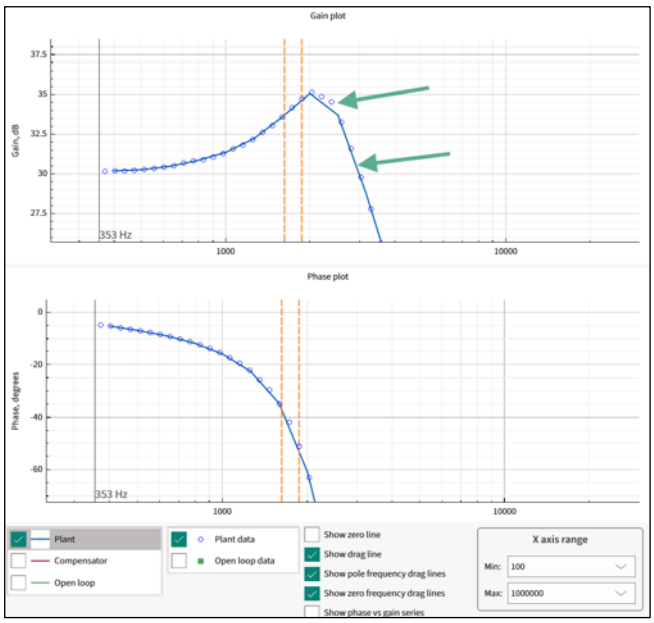


7.6.9 Step 9

To promote Plant poles, zeros, and gain with the calculated values, double-click the left arrow button next to the Plant poles. Now, TF panes for **Plant** and **Open loop** are populated because the Tool has both Compensator and Plant poles and zeros.



The **Bode** plot zoom view of the extracted plant data points and the line calculated from the identified plant poles and zeros.



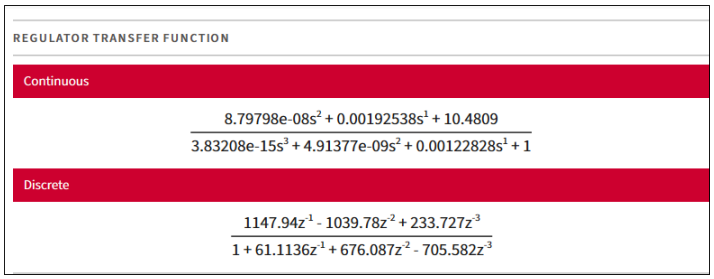
Note: Imported Open loop and Plant data is available only in the runtime. Plant poles and zeros are saved in the .mtbcoef file alongside with the compensator parameters.

8 Regulator Transfer Function pane (Compensator tab)

The **Regulator Transfer Function** pane displays the **Continuous** and **Discrete** formulae), **Float-point numbers,** and **Fixed-point 24-bit for HW** panes.

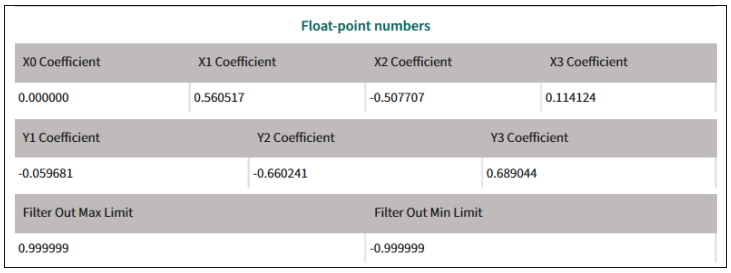
8.1 Regulator TFs

Every time the poles or zeros are updated, **Continuous** and **Discrete** Transfer Functions are updated to reflect the change.

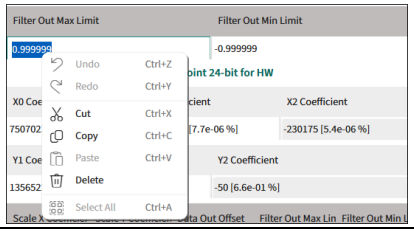


Respective Discrete coefficients are listed in the **Float-point numbers** table.

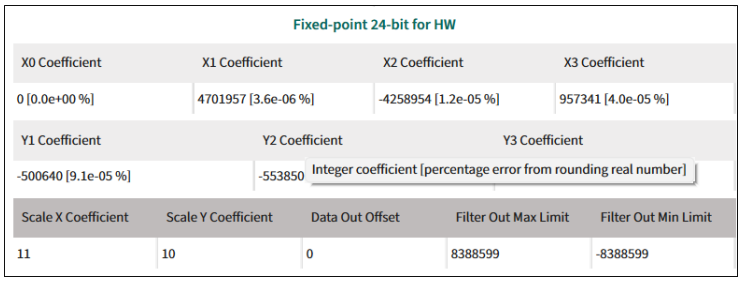
8.2 Float-point numbers



These values correspond to the normalized values to be passed to the Device Configurator:

Parameter	Description
X0 Coefficient	Scaled product on **X0 Coefficient**
X1 Coefficient	Scaled product of on **X1 Coefficient**
X2 Coefficient	Scaled product of on **X2 Coefficient**
X3 Coefficient	Scaled product of on **X3 Coefficient**
Y1 Coefficient	Scaled Y1 filter coefficient
Y2 Coefficient	Scaled Y2 filter coefficient
Y2 Coefficient	Scaled Y3 filter coefficient
Filter Out Max Limit	The filter out maximum/minimum limit register in a float-point representation. Double-click the value to edit. Right-click the value to do some action. 
Filter Out Min Limit	

8.3 Fixed-point 24-bit for HW



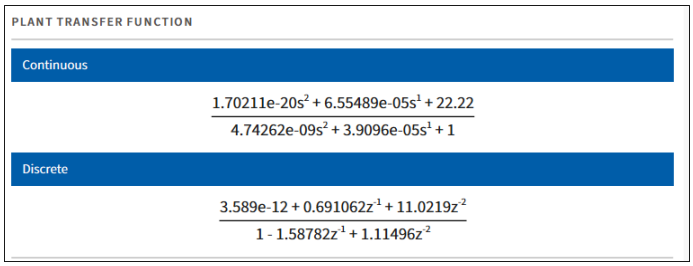
The resulting list of values is required to initialize an HW3P3Z instance.

Parameter	Description
X0 Coefficient	Scaled product on **X0 Coefficient** [percentage error] The integer coefficient. The relative error in %, introduced by discretization from the float-point number to integer number.
X1 Coefficient	Scaled product of on **X1 Coefficient**
X2 Coefficient	Scaled product of on **X2 Coefficient**
X3 Coefficient	Scaled product of on **X3 Coefficient**
Y1 Coefficient	Scaled Y1 filter coefficient
Y2 Coefficient	Scaled Y2 filter coefficient
Y2 Coefficient	Scaled Y2 filter coefficient
SCALE X Coefficient	Scale X coefficients factor register
SCALE Y Coefficient	Scale Y coefficients factor register
Data Out Offset	Filter offset register
Filter Out Max Limit	Filter out maximum limit register
Filter Out Min Limit	Filter out minimum limit register
G_OUT	Output scale down factor

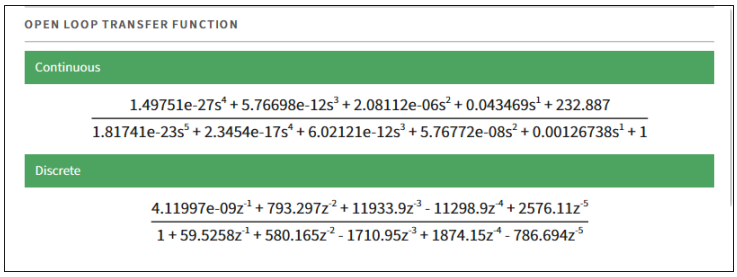
9 Plant Transfer Function pane

The **Plant** tab displays the **Plant Coefficients** and **Open Loop Coefficients** panes.

9.1 Plant TFs



9.2 Open Loop TFs



10 Coefficients generation flow

This section describes the case when the Filter Coefficient Tool is launched outside the Modus Toolbox™.

10.1 STEP 1

Open the Filter Coefficient Tool and click **New**.

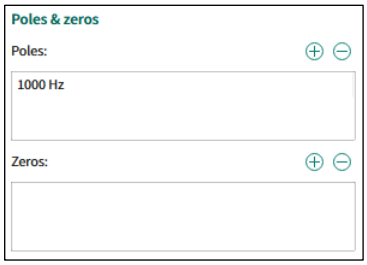
Add the Poles and Zeros needed by the target application. For simplicity, create a single pole, Low Pass Filter with cutoff frequency 1 kHz. Note that the implementation considers a pole written in the following format:

$$\frac{1}{s + (2 \cdot \pi \cdot f_{POLE})}$$

Similarly, Zero follows the format:

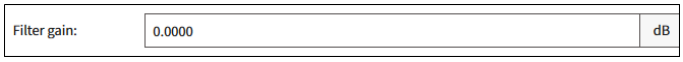
$$s + (2 \cdot \pi \cdot f_{ZERO})$$

But the table shows only the following:



10.2 STEP 2

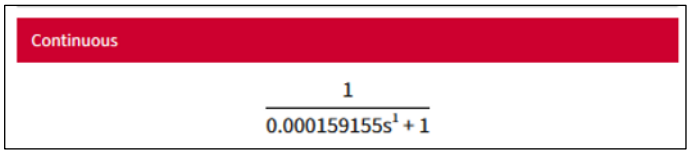
Adjust the filter gain. The LPF as a unity gain for frequencies below the cutoff frequency – for our case below 1kHz. Therefore, it is required to enter a gain value that matches the expected LPF behavior in the low frequency range, for example, at 0 Hz, the expected gain is 1 or 0 dB.



The formula behind the gain calculation: $(2 \cdot \pi \cdot f_{POLE})$

10.3 STEP 3

The **Continuous Transfer Function** displays:



The numbers representation rounding might affect the theoretically expected equation.

10.4 STEP 4

Define the discretization time step to transform the **Continuous Transfer Function** into its Discrete equivalent. The Discrete time step represents the execution speed of the filter described by Discrete Transfer Function. Here, a 100 kHz update rate is applied equivalent to 1e-5 sec.

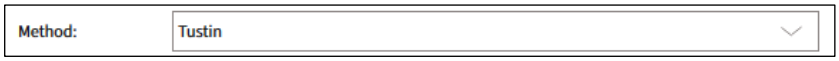


10.5 STEP 5

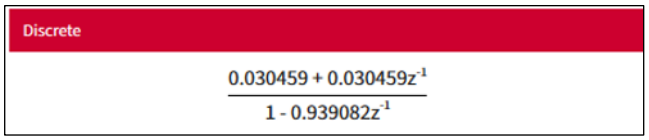
Select the discretization method to transform Continuous Transfer Function into its equivalent Discrete form. The following methods are available:

- Zero order hold
- First order hold
- Tustin
- Backward difference
- Forward difference

For LPF, the default **Tustin** method is selected.



The final Discrete Transfer Function is now generated.



10.6 STEP 6 (Start of coefficient manipulation to fit into HW3P3Z design)

From Discrete Transfer Function, the coefficients are extrapolated to be adjusted to a format acceptable by the HW3P3Z filter. The first step consists of converting the coefficients into equivalent float-point values.

If the Filter Coefficient Tool is opened in stand-alone mode, such float-point values can be copied and pasted into the Device Configurator:

- for **X#** coefficients, as they are
- for **Y#** coefficients, with a reversed sign.

10.6.1 Numerator coefficients

1. From **Step 1**, coefficients are normalized to be in the range **[-1, 1)**.
   Normalization automatically determines n between 0 and 7 and applies the 2^n scaling (numerically equivalent to an "n-"bit right shift). The identified n will be presented in the Scale X Coefficient field. Mathematically, it is calculated as X#/2^n.
2. For the LPF, Scale X Coefficient = 0.

   The expected coefficient range is between -1 and +1. When **X** coefficient values are getting closer to 0, their equivalent representation in the 23-bit fractional representation loses resolution.
   The calculated integer values to write into the HWFILT register will be simply:
   X#_integer = X#_normalized * *2^23 – 1*.
3. The user can decide on adding attenuation on the output of the filter through the G_OUT factor. Every LSB defined in this field will be responsible to attenuate by -6dB the resulting filter design. The max attenuation is -42dB since G_OUT range is 0-7.



10.6.2 Denominator coefficients

1. Denominator **Y#** coefficients are normalized to be in the range *[-1, 1)*.
   The normalization procedure is equivalent to that described for **X#** and it ends defining the **Scale Y Coefficient** field.
   For the LPF, *Scale Y Coefficient = 1*.
2. Then, denominator coefficients signs are inverted. This happens when the Transfer Function (written in the normalized form as represented in the Configurator) is transferred to its corresponding difference equation. Then, the denominator coefficients are moved to the right side of the equation.

Y1 Coefficient
0.939082

Direct Form I representation:

$$H(z) = \frac{X0 + X1z^{-1} + X2z^{-2} + X3z^{-3}}{1 + Y1z^{-1} + Y2z^{-2} + Y3z^{-3}}$$

Transformed into a difference equation representation:

$$y[k] = X0 \cdot x[k] + X1 \cdot x[k-1] + X2 \cdot x[k-2] + X3 \cdot x[k-3] - Y1 \cdot y[k-1] - Y2 \cdot y[k-2] - Y3 \cdot y[k-3]$$

10.7 STEP 7

Generation of the integer numbers used to initialize or update the HW3P3Z registers during the run time. To determine their values, the float-point numbers, both **X#** and **Y#** in their final scaled and normalized value, are multiplied by factor *2^23 – 1*. Hence, the coefficients will be represented in the fixed 24-bit format.

Fixed-point 24-bit for HW	
X0 Coefficient	X1 Coefficient
255509 [1.6e-04 %]	255509 [1.6e-04 %]
Y1 Coefficient	
7877590 [2.0e-06 %]	

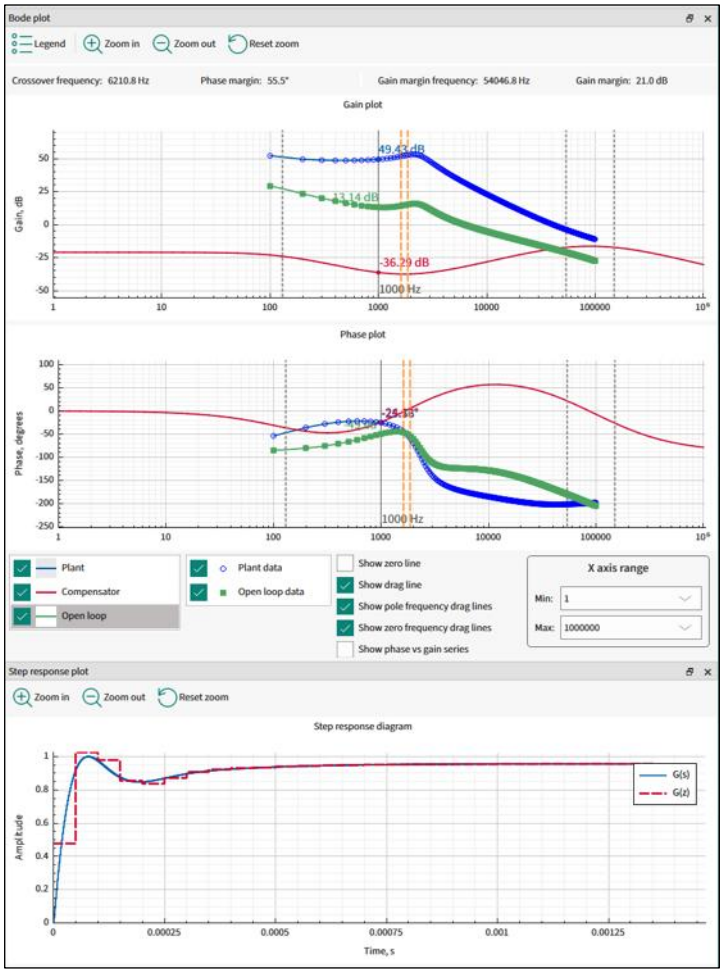
10.8 STEP 8

To complete the configuration of the HW3P3Z filter, define three more fields: **Data Out Offset**, **Filter Out Max Limit**, and **Filter Out Min Limit**. Additionally, for PSC3P6 or PSC3M6 MCU, the Anti Windup Limits (refer to section 6.6) are required to be defined as well. These fields are editable and must be specified by the user.

For LPF, implementation **Data Out Offset** is set to 0, while **Filter Out Max Limit** and **Filter Out Min Limit** are left set to 99% of the available range, calculated as **ABS***(0.99 * 2^23 – 1)*.

11 Plots (graphs)

The **Bode plot** and **Step response plot** comprise a set of graphs that allow assessing a given power conversion design.



Bode plot

The graphs are **Gain plot** (magnitude) and **Phase plot** of a Transfer Function versus frequency. The gain expressed in decibels and phase in degrees are plotted versus frequency using semi-logarithmic axes. A plot set consists of the **Plant**, **Compensator**, and **Open loop series**. Imported data is built when it is available as scatter points.

Step response plot

The graph displays the time-domain response of the closed-loop system to a unit step input. The step response plot can be calculated only if the Plant model is available. The amplitude (output value) is plotted versus time using linear axes. The plot displays how the system output evolves from its initial state to its final steady-state value, revealing key performance characteristics such as rise time, settling time, overshoot, and steady-state error. A plot set consists of both Continuous-time G(s) and Discrete-time G(z) closed-loop step responses, calculated from the compensator and Plant Transfer functions with unity negative feedback.

Plots (graphs)

11.1 Toolbar



The toolbar consists of the **Hide**/**Show** legend; **Zoom in**; **Zoom out**; Reset zoom commands. You can also zoom or move a graph using the mouse. With additional modifiers, you can zoom only the **X** axis (**Shift + Wheel**) or **Y** axis (**Ctrl + Wheel**).

11.2 Margins bar

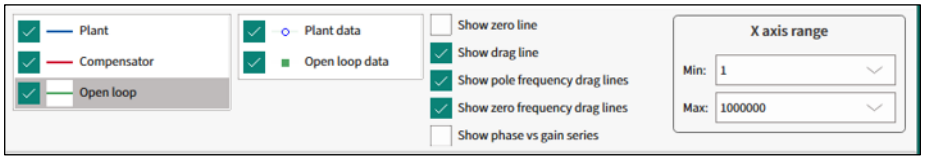


The margins bar displays the open-loop stability metrics:

- **Crossover frequency (Hz)** – The frequency where the open-loop gain reaches 0 dB.
- **Phase margin (°)** – The difference between the open-loop phase at the gain crossover and −180°.
- **Gain margin frequency (Hz)** – The frequency where the open-loop phase reaches −180°.
- **Gain margin (dB)** – The difference from the open-loop gain at the −180° phase crossover to 0 dB.

A value is shown as **N/A** when the corresponding crossover cannot be determined.

11.3 Legend



The **Legend** provides an explanation of displayed graph data and allows for the customization of this information. Select the relevant option in the **Legend** to display the various subsets of series.

The following options are available:

- **Show zero line** – Reveals a vertical static line that indicates the **X** coordinate where the open loop series intersects with the 0 dB **Y**-axis value, as well as the **Phase margin**.
- **Show drag line** – Displays a vertical draggable line that presents the values of the bode plot series at their intersection with the line at a certain frequency. You can move this line using the mouse. If it is beyond the **X** axis range, double-click the chart area to make it visible.
- **Show pole/zero frequency drag lines** – Displays vertical draggable lines on the **Bode plot** at the current pole frequency for the active compensator. Drag these lines horizontally to interactively adjust pole positions in Hz – the compensator Transfer Function updates in real-time. Labels display during dragging showing the current frequency value. Click and drag any vertical pole/zero line left or right to change its frequency. Release the mouse to apply the new value to the active compensator configuration.
- **Show phase vs gain series** – Displays the phase series on the **Gain plot**. The label "Phase" will be included into the legend for differentiation.
- **X-axis range** – Remains consistent for both **Gain** and **Phase** plots and can be modified by selecting the desired min and max values from the dropdown menus. The widest available range spans from 0 Hz to 1 MHz. The **Y**-axis range adjusts dynamically to ensure the visualization of all series.

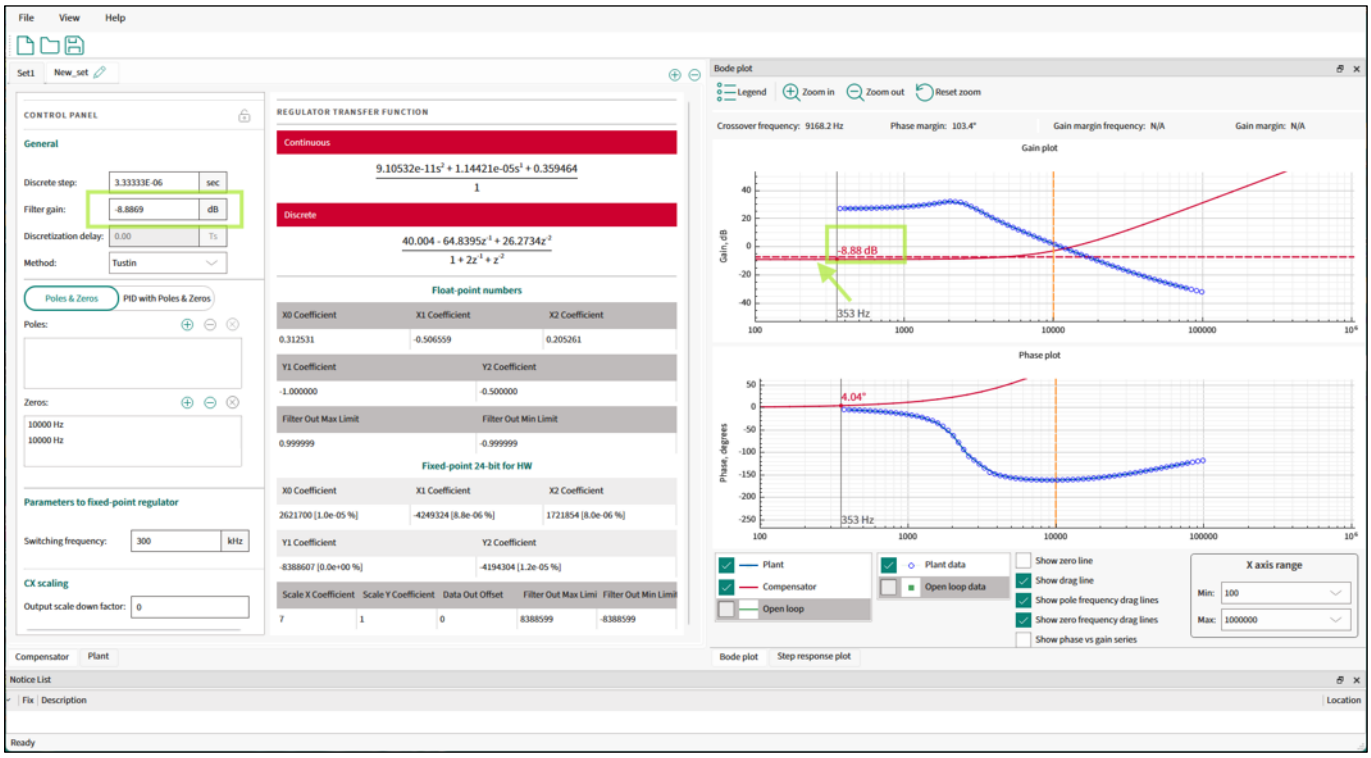
Plots (graphs)

- **Compensator Curve Drag Against Gain Axis** – The compensator gain curve drag-and-drop shifting against **Gain** axis is available. In the **Gain plot**, click directly on the Compensator curve (a helper horizontal orange line displays to track the current mouse position) and drag vertically up or down. The compensator gain changes proportionally to mouse movement. Upon the mouse release, new gain value are submitted as Compensator gain to the active set and trigger recalculations of all dependencies.

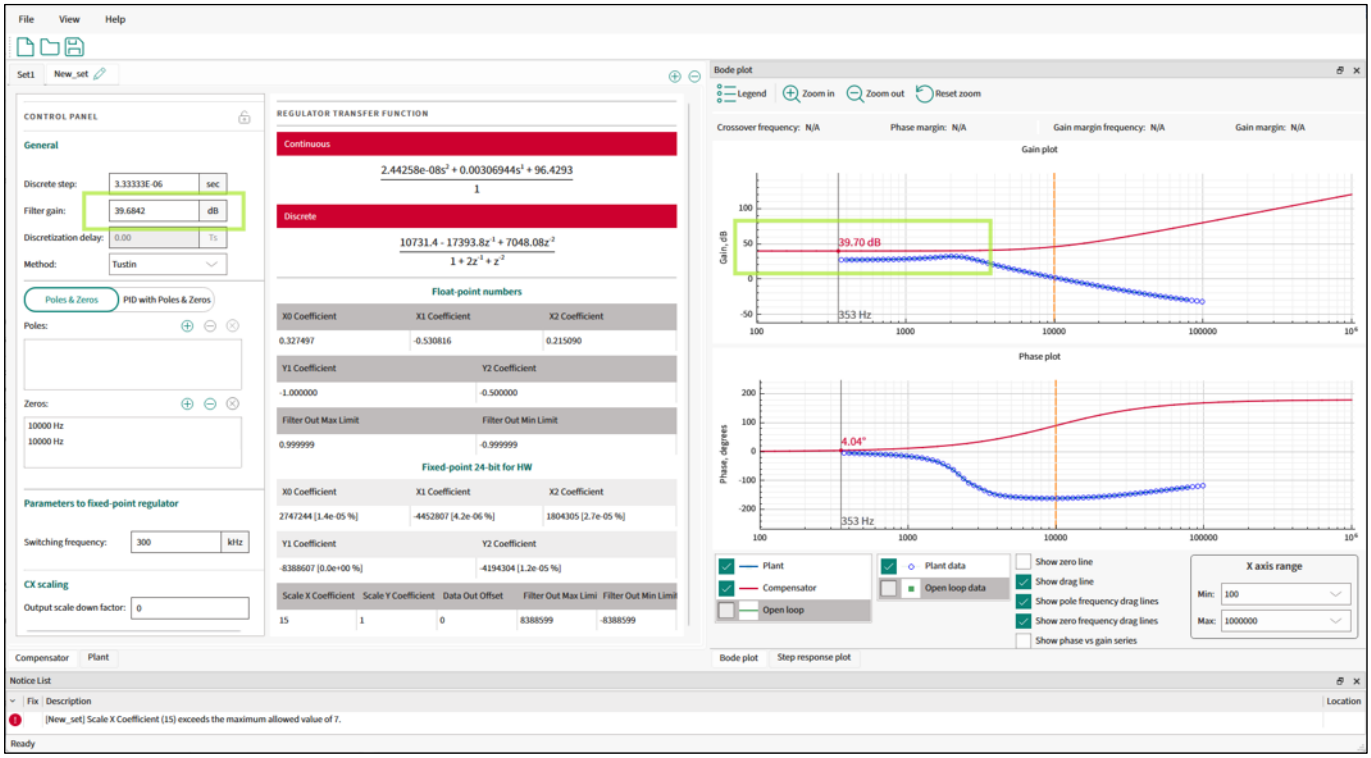
12 Flow for compensator fast design via bode plot manipulations

12.1 Step 1 (Compensator gain adjustment)

Go to the **Compensator** tab, click and hold the left mouse button on the Compensator curve. The horizontal dashed line displays. Drag the dashed line to a new gain position at 40 dB and release the mouse button.

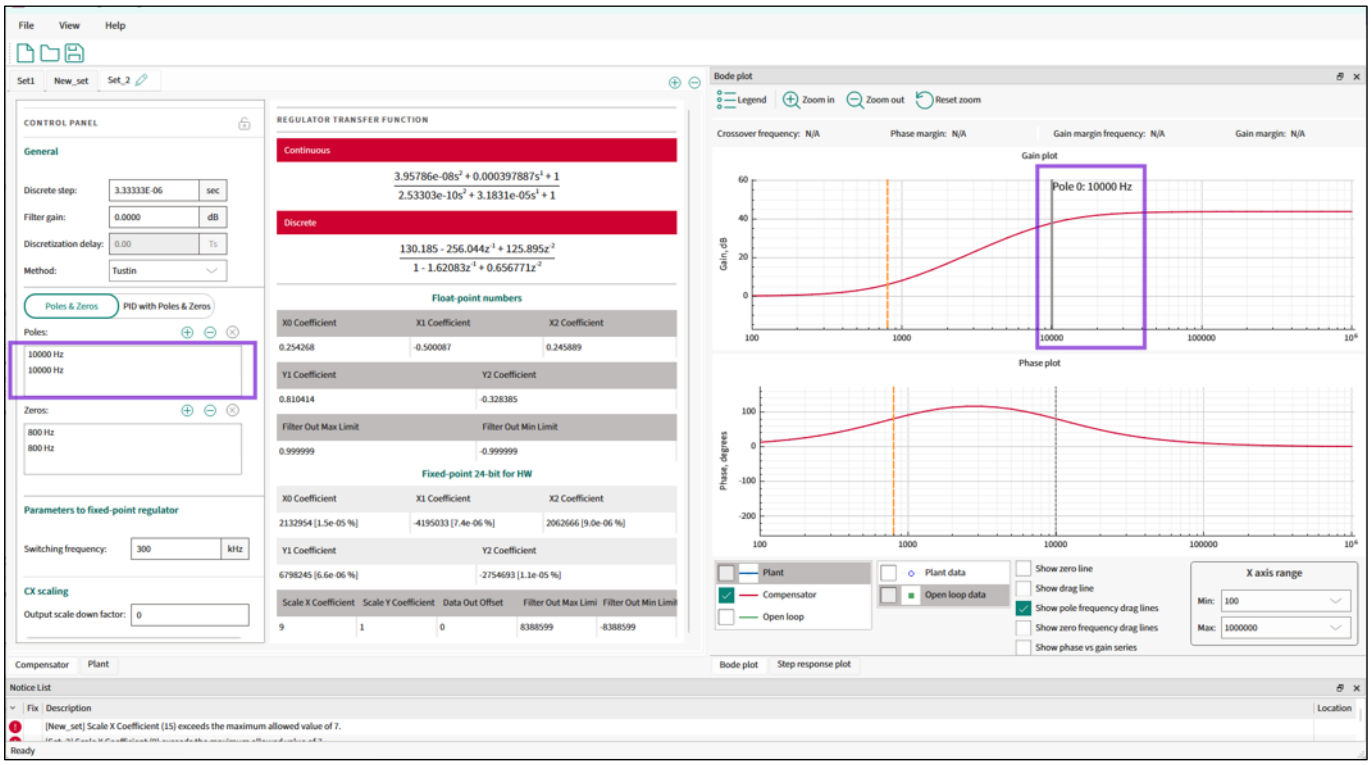


The whole Compensator curve is shifted vertically to match the new gain at the selected frequency. The Compensator gain value is updated accordingly.



12.2 Step 2 (Compensator poles adjustment)

On **Bode** plot legend, enable the **Show pole frequency** drag line. Click and hold the left mouse button on the pole frequency drag line. On the top of the line, observe the selected pole and its current position. Drag the line to a new position and release the mouse button.

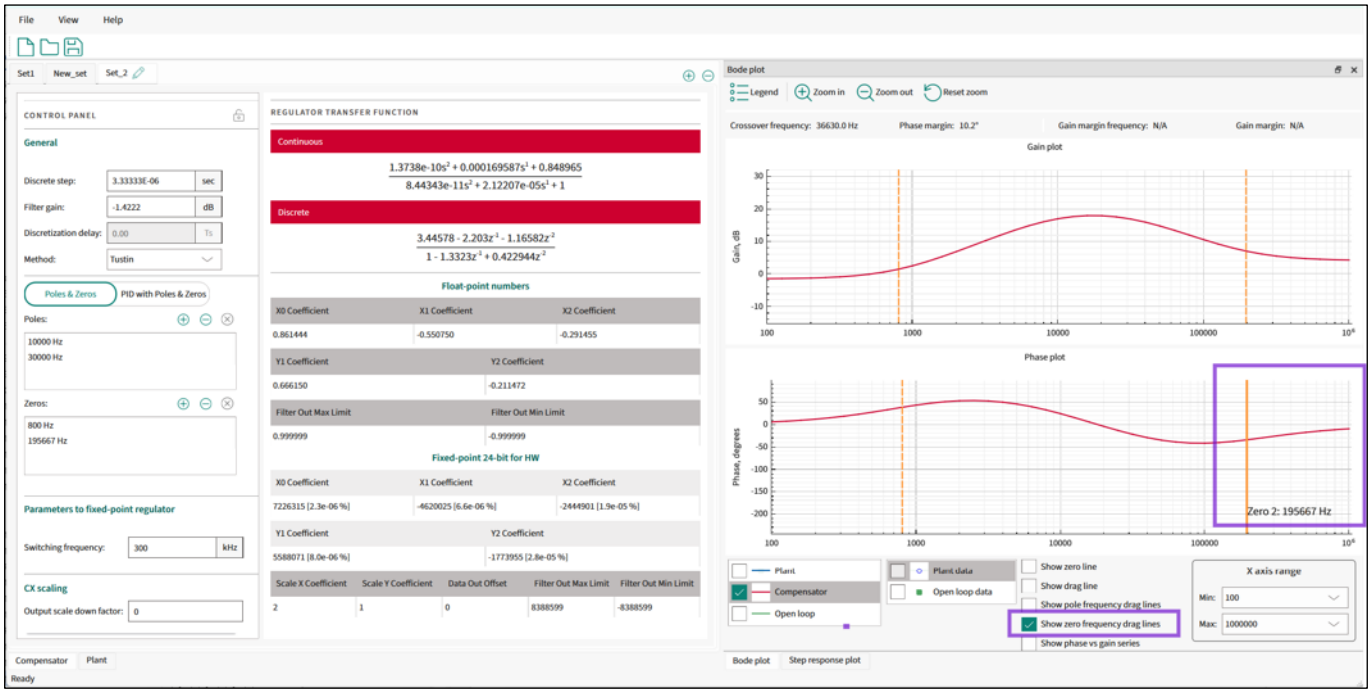


The **Bode** plot and **Poles & Zeros** pane are updated with the new pole frequency value.



12.3 Step 3 (Compensator zeros adjustment)

On the **Bode** plot legend pane, enable the **Show zero frequency** drag line. Click and hold the left mouse button on the zero frequency drag line. At the bottom of the line, observe the selected zero and its current position. Drag the line to a new position and release the mouse button. The **Bode** plot and **Poles & Zeros** pane are updated with the new zero frequency value.



13 Version changes

This section lists and describes the changes for each version of this tool.

Version	Change descriptions
1.0	New tool.
1.1	Added Bode plot.
	Added Output scale down factor.
	Updated tables headers and content.
	Updated the Filter gain ranges.
1.1.1	New: • Extended the range for the Discrete Step and PWM frequency • Made non-editable the values in the coefficient tables
	Stability fixes: • scaling for Y coefficients • saving the pole at origin in the **.mtbcoef* file • calculating Discrete TF for all supported discretization methods.
1.2.0	Extended the Float-point numbers table.
	Added validation messages in the Notice List.
	Stability fixes.
1.2.1	Added "Paste suggested filter gain" button.
	Added dynamically-calculated "percentage error from rounding real number" to **k_OUT x (X0 Coefficient)** on "Fixed-point 24-bit for HW" table.
	Performance and stability fixes
1.3.0	Gain unit migration from decimals to dB.
	Added drop-down with supported MCUs.
	Added Plant tab, with complex poles and zeros, Plant and Open loop, Transfer function formulas.
	Added Plant experimental bode plot data import and transfer function identification.
	Added Bode plot with Compensator, Plant and Open loop curves.
	Added Step response plot.
	Reduced the number of parameters in the "Parameters to fixed-point regulator" block.
	Fixed bugs.
1.4.0	Added: • lock button to freeze Mother set • PID with Poles & Zeros input mode • complex poles and zeros support for Compensator tab • anti-windup parameters • import for open loop data file • Plant data extraction from open loop data • loaded data preprocessing • margins with open-loop stability metrics to Bode plot • legend for plant and open loop data on Bode plot
	Extended Bode plot X-axis max range to 10 MHz.
	Fixed bugs.

Revision history

Revision	Date	Description
**	2025-08-19	New document.
*A	2025-10-24	Updated to version 1.1. Added section "Coefficients generation flow".
*B	2025-11-07	Updated to version 1.1.1.
*C	2025-12-11	Updated to version 1.2.0.
*D	2026-04-03	Updated to version 1.2.1. Renamed to "Power Suite Filter Coefficient Tool".
*E	2026-05-26	Updated for ModusToolbox™ PSOC Control C3 M6 EAP version 0.1.0.
*F	2026-08-17	Updated to version 1.3.0.
*G	2026-10-27	Updated to version 1.4.0. ModusToolbox™ Power Suite 1.2.0.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2026-10-27

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2026 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

002-41662 Rev. *G

Important notice
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”)

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.