

### Key new features in SmartCtrl 2024.1

New AC-DC converter topology: **Three-Phase PFC Boost Converter**.

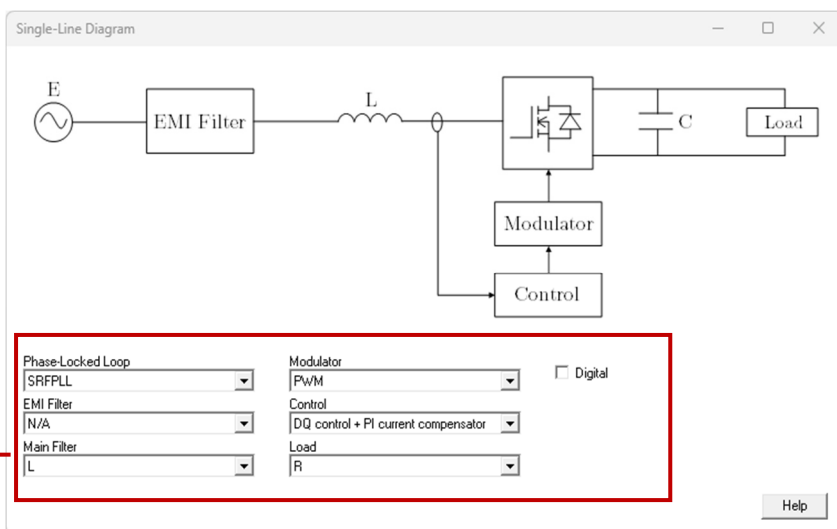
- Two different control structures are fully supported:
  - Alpha-Beta control
  - DQ control
- Two main filter types can be selected:
  - L Filter
  - LCL Filter
- In the case of the LCL Filter, active damping or passive damping can be chosen.
- Proportional-Resonant compensator can be tuned for alpha-beta control structure.
- The control loop of two different Phase-Locked Loop (PLL) can be tuned:
  - Synchronous Reference Frame PLL (SRFPLL)
  - Quadrature Signal Generator - Synchronous Reference Frame PLL (QSG-SRFPLL)
- Four widely used EMI Filter structures have been implemented.
- A new way of defining the sensing chain is included.
- Five types of modulators can be selected.
- System level stability analysis tools has been developed. Output Impedance of EMI filter vs Input Impedance of the controlled converter.
- Designs can implement s-domain and full digital control loops.
- Independent floating windows to enhance user experience.
- Export to third-party simulators.
- Automated C-code generation.

## New AC-DC converter topology: Three-Phase PFC Boost Converter

The **Three-phase PFC Boost Converter** is a bidirectional two-level rectifier. This three-phase rectifier is frequently used in applications such as electric vehicle (EV) chargers, grid energy storage systems and uninterruptible power supplies (UPS), motor drives and telecommunication power supplies.

The **single line diagram** window allows the user to easily configure the type of control structure, phase-locked loop, EMI Filter, modulator, main filter and load.

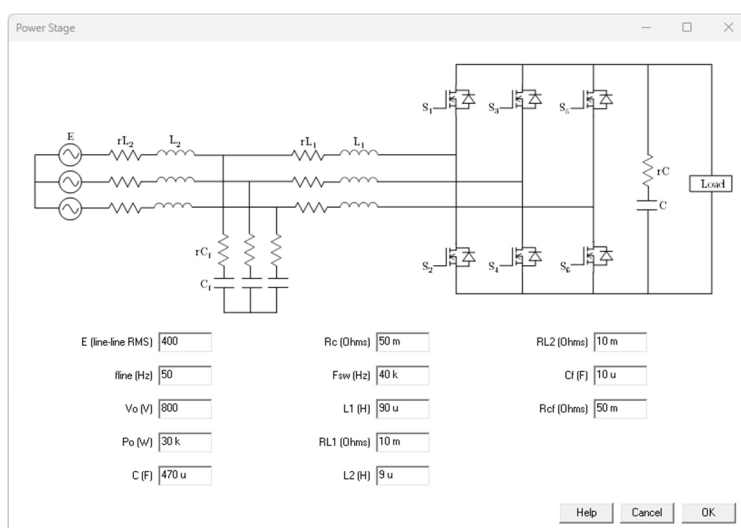
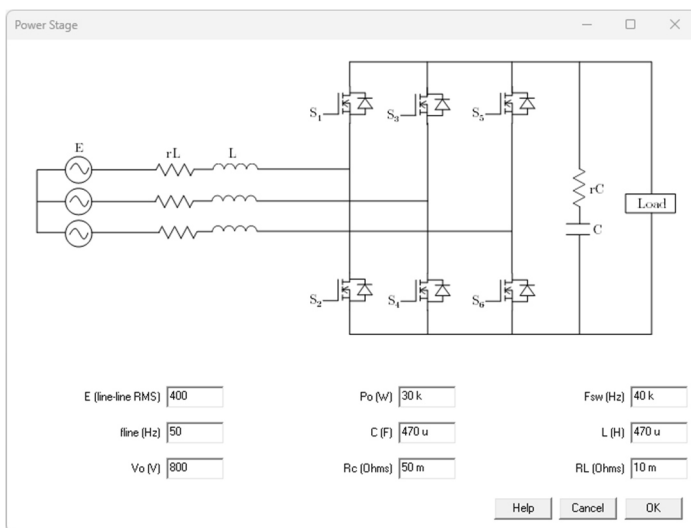
Just selecting the digital option, the user is able to design control loops in z-domain.



**SmartCtrl 2024.1** allows to design the control loops for different types of main filters in a simple way.

L Filter

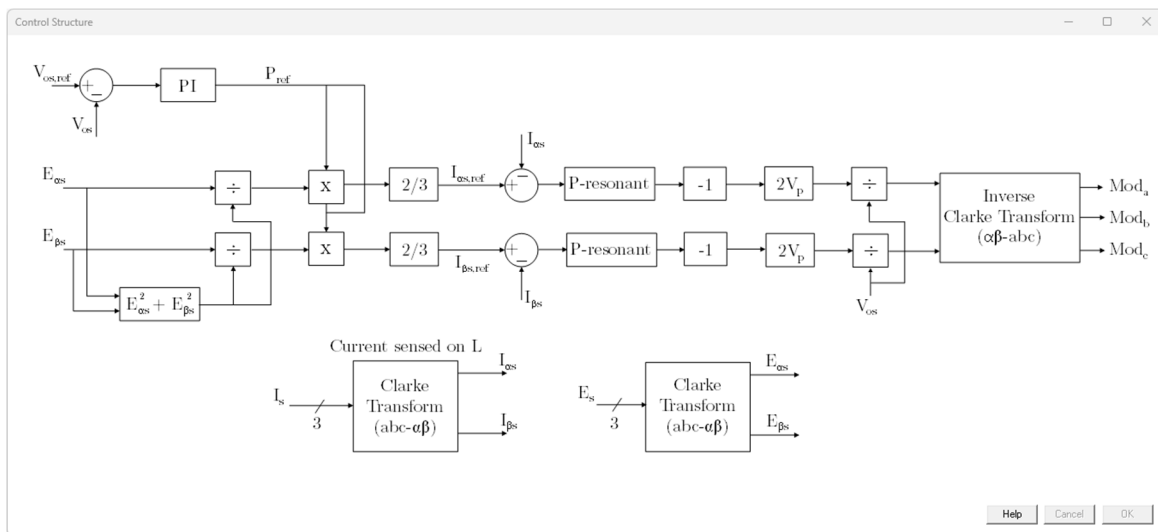
LCL Filter



### Designing the control loops of the most used control structures in three-phase rectifiers:

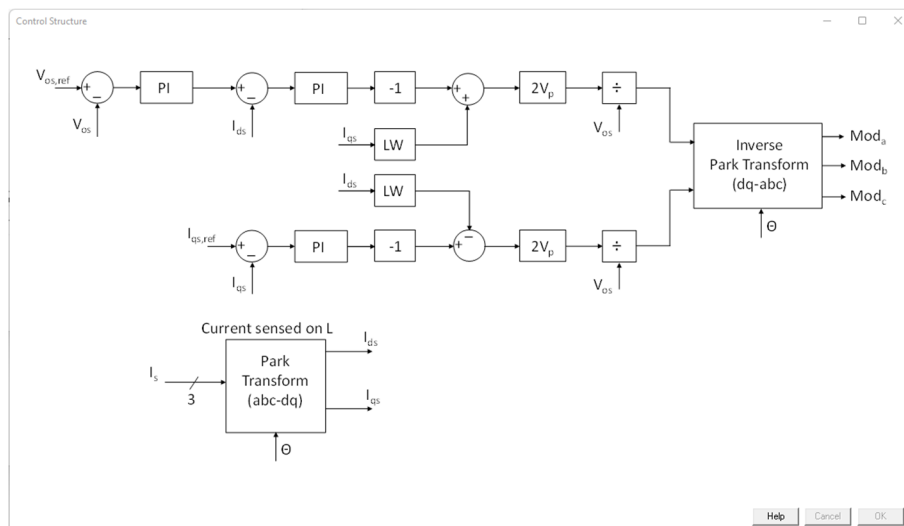
#### Stationary Reference Frame or Alpha-Beta control

This control structure may be seen easy to implement, however active and reactive power are coupled. The inner loop current reference generator is designed to impose zero reactive power; therefore, the external power loop can control the active power of the converter.



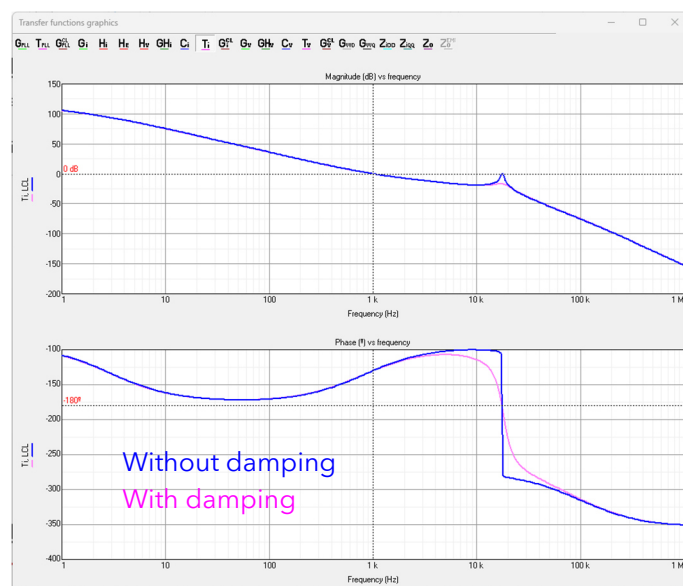
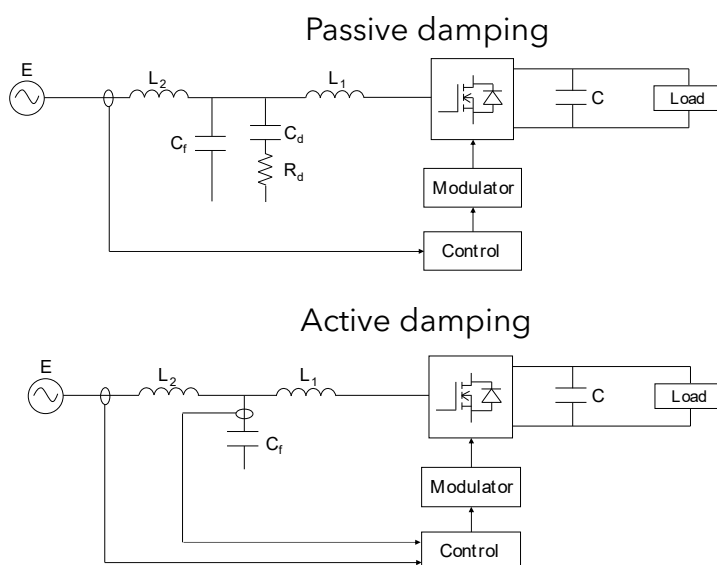
#### Synchronous Reference Frame or DQ control

This control structure offers a key benefit by converting time-varying three-phase signals into DC signals. Another significant advantage lies in the decoupling of active and reactive power. As a result, the d-axis is utilized for controlling active power, while the q-axis is employed for managing reactive power, or vice versa. **SmartCtrl 2024.1** allows to design the control loops considering an alignment with the d-axis or q-axis.



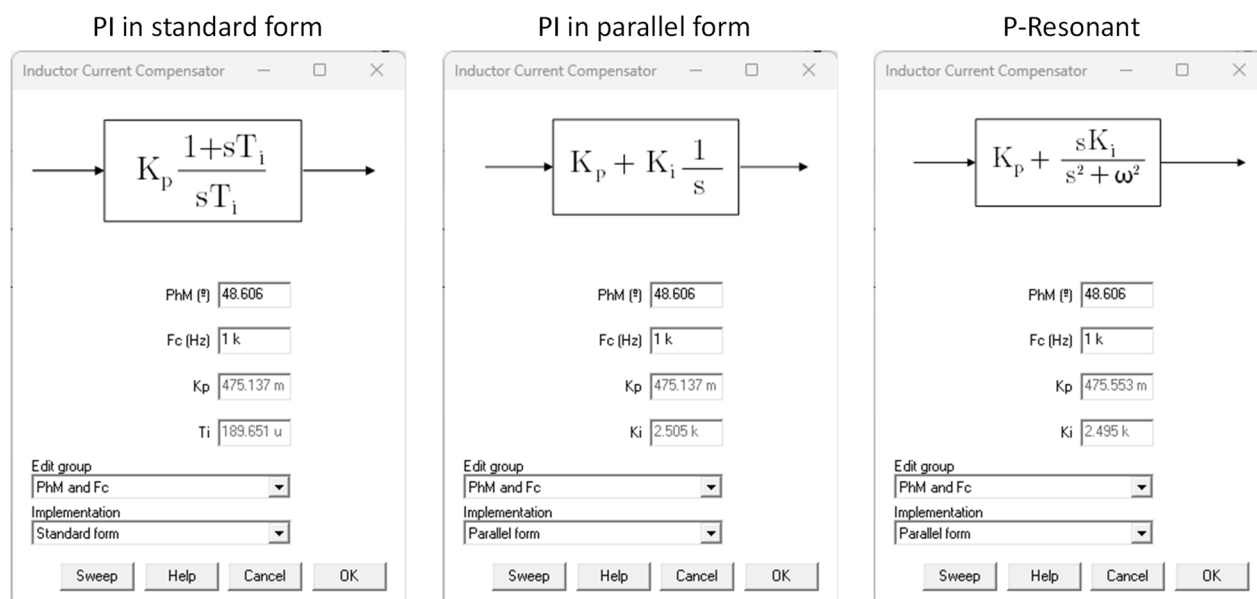
## LCL Filter Challenge

In case of LCL filter, one of the main challenges when designing the control loops is the resonance peak that the grid-side inductor current plant has; for this reason, **SmartCtrl 2024.1** allows to design the control loops considering active and passive damping techniques in order to ensure the stability of the system.



## Different Types of compensators

In **SmartCtrl 2024.1**, the user can choose different implementations of the Proportional-Integral (PI) compensator. On top of that, for Alpha-Beta control, **Proportional-Resonant compensators** can be tuned.

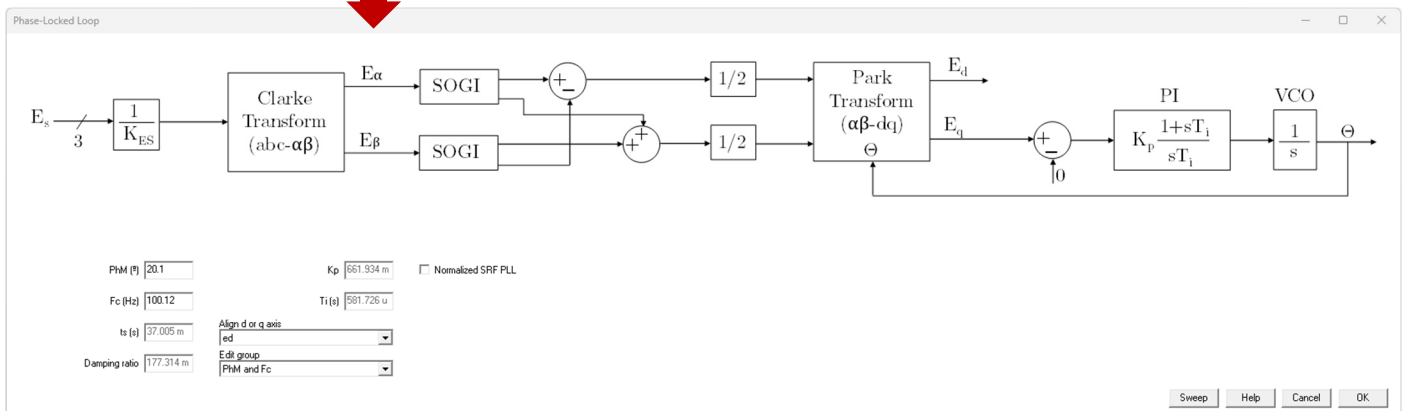
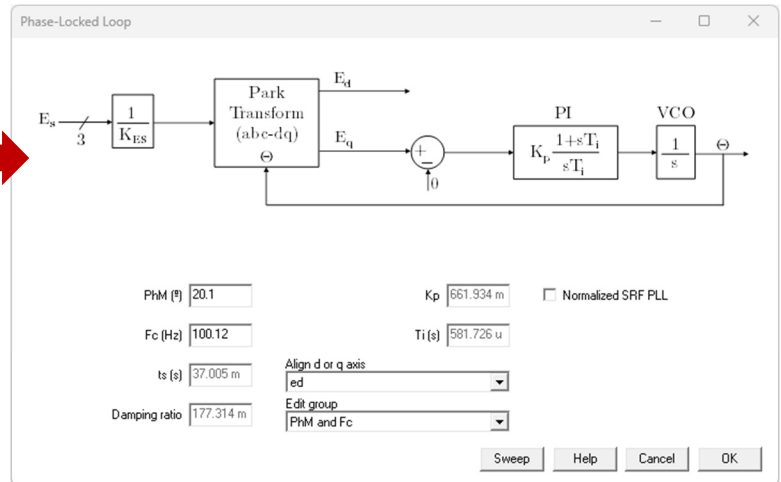


### Phase-Locked Loop

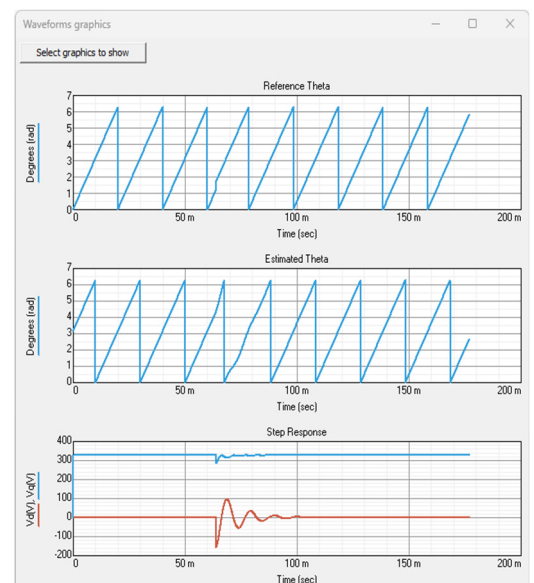
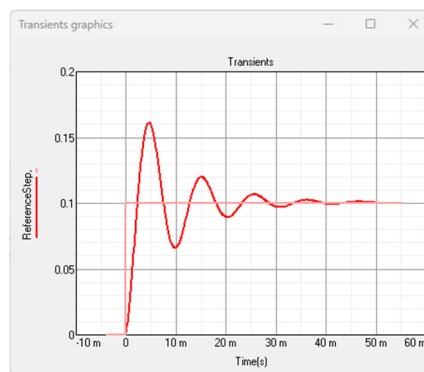
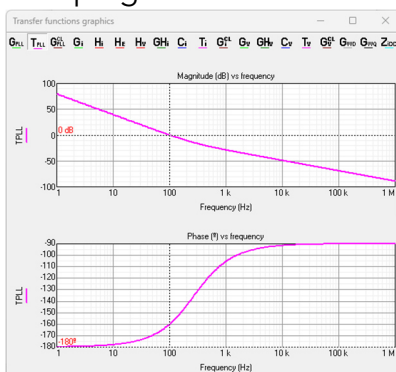
In DQ controls, a Phase-Locked Loop (PLL) that detects the phase angle of grid voltage is widely used. **SmartCtrl 2024.1** allows the user to design the PLL control loop.

Two types of PLLs can be designed:

- **SRFPLL** (Synchronous Reference Frame Phase-Locked Loop)
- **QSG-SRFPLL** (Quadrature Signal Generator - Synchronous Reference Frame Phase-Locked Loop). The Quadrature Signal Generator is used to detect positive or negative sequence.

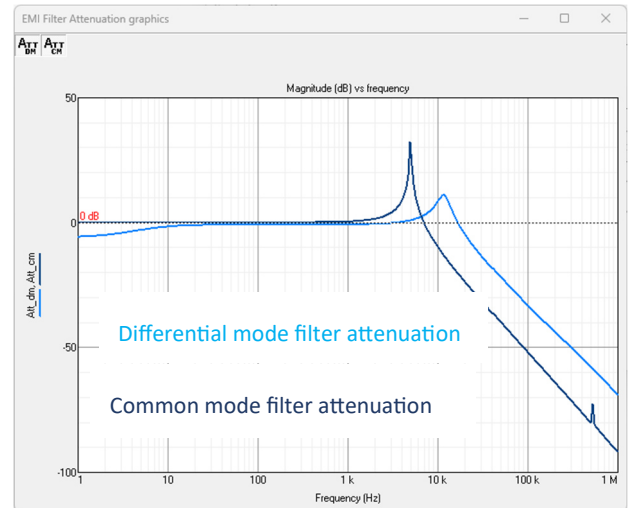
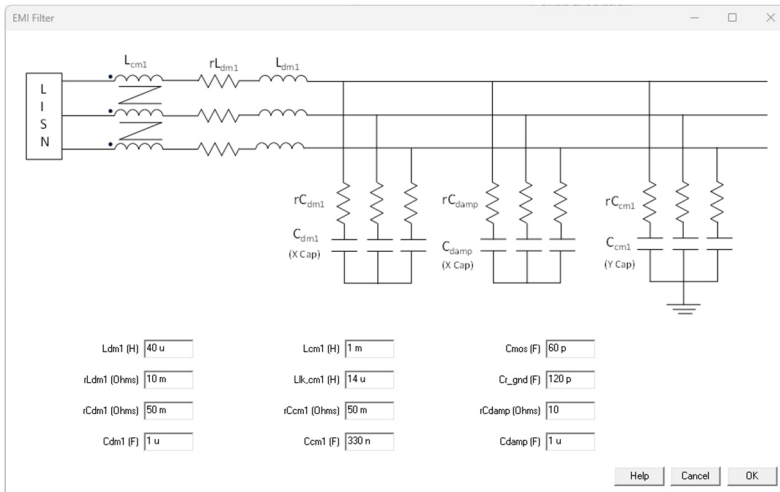


**SmartCtrl 2024.1** provides the capability for users to observe the transient response of the system by introducing a phase step into the grid voltage angle. This feature facilitates the analysis of stabilization time and damping ratio.



## EMI Filter

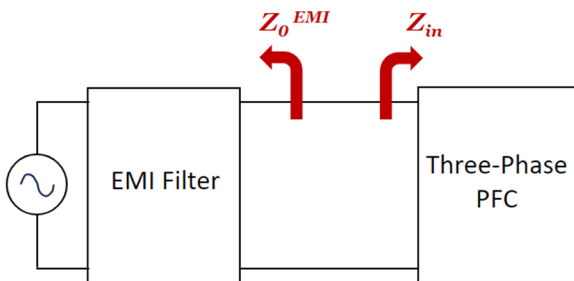
**SmartCtrl 2024.1** allows the user to choose among four predefined EMI filter structures. By varying the EMI filter parameters, the **attenuation** achieved by the **common-mode** and **differential-mode** filter can be observed. This tool allows the user to obtain a **Preliminary EMI filter design**.



## System Level Stability Analysis

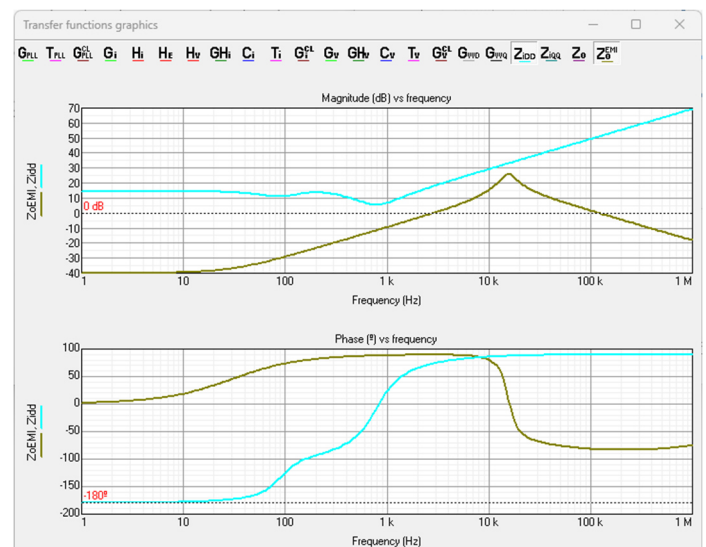
**SmartCtrl 2024.1** also allows to predict the system stability when the EMI filter is connected. The system level stability is analyzed using the **impedance criterion** (Middlebrook & The Gain Margin and Phase Margin Criterion). **Two impedances are calculated:**

- EMI filter output impedance.
- Input impedances in the dq frame of the controlled converter, including the main filter and current and voltage loops.



Middlebrook Criterion:  $\left\| \frac{Z_o^{EMI}}{Z_i} \right\| \ll 1$

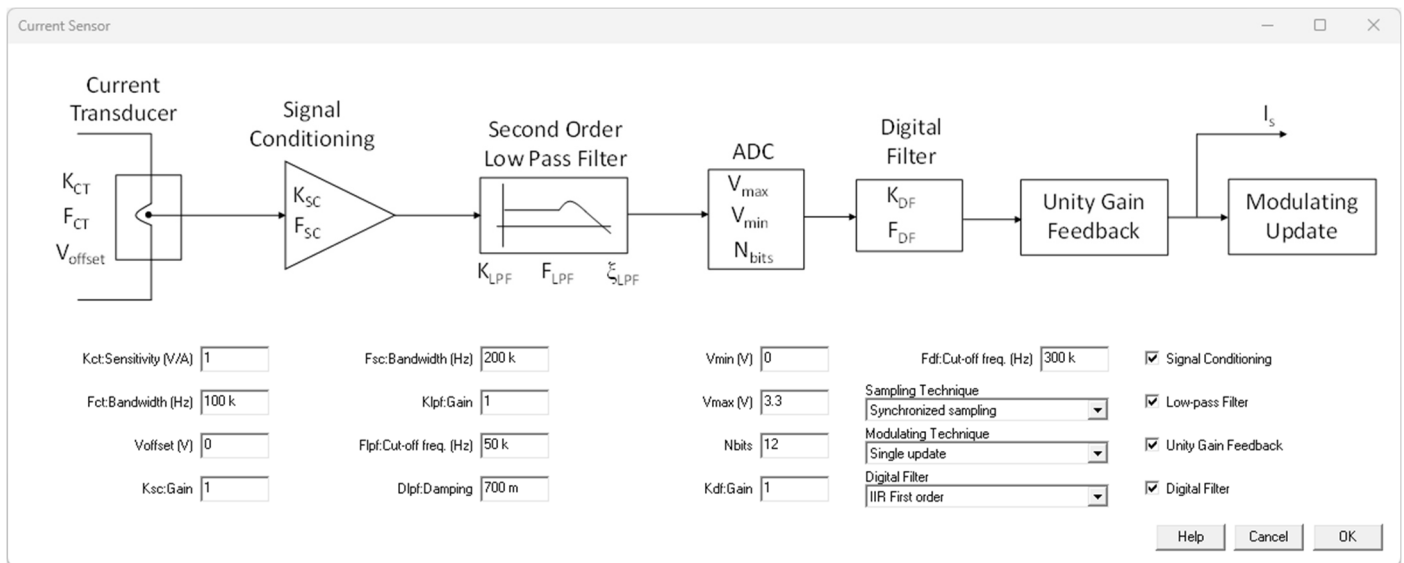
The Gain Margin and Phase Margin Criterion:  $\left\| \frac{Z_o}{Z_i} \right\| \leq \frac{1}{GM}$  and  $|\arg(Z_o) - \arg(Z_i)| \leq 180^\circ - PM$



### A new way of defining the sensing chain

In **SmartCtrl 2024.1**, the sensing chain can include conditioning stage, antialiasing filter, and digital filters. The **digital filters** that the user can choose are: **IIR First order**, **IIR Second order** and **FIR Moving Average Filter**.

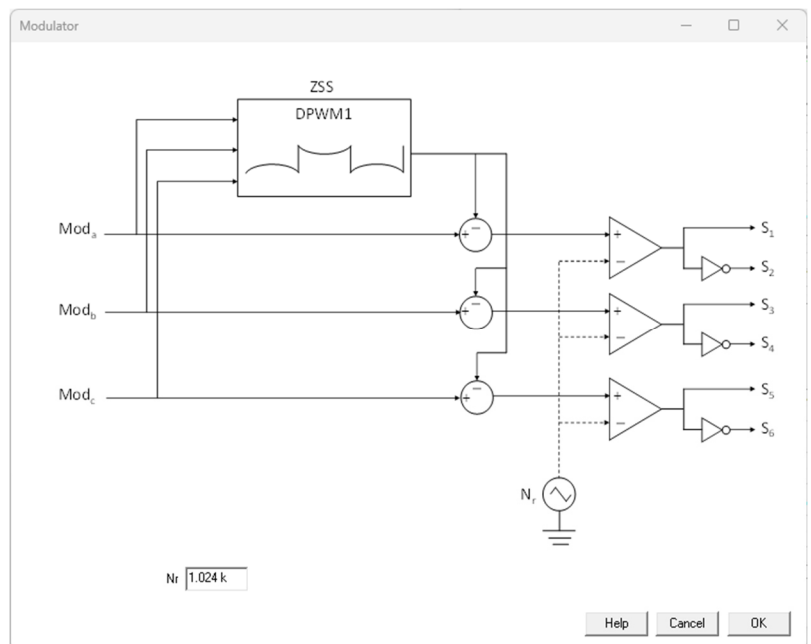
The **Unity Gain Feedback** option allows the user to know the **total inverse gain** (the gain to compensate the sensing stage) that must be implemented in the control structure in order for the sensed current or voltage to be equal to the actual value.



### Modulators

The user can choose **five types of modulators**, which are:

- **PWM**: Sinusoidal Pulse Width Modulation.
- **SVPWM**: Space Vector Pulse Width Modulation.
- **Discontinuous 1**: 60-degree Discontinuous Pulse Width with positive and negative DC clamping.
- **Discontinuous Max**: 120-degree Discontinuous Pulse Width Modulation with positive DC clamping.
- **Discontinuous Min**: 120-degree Discontinuous Pulse Width Modulation with negative DC clamping.

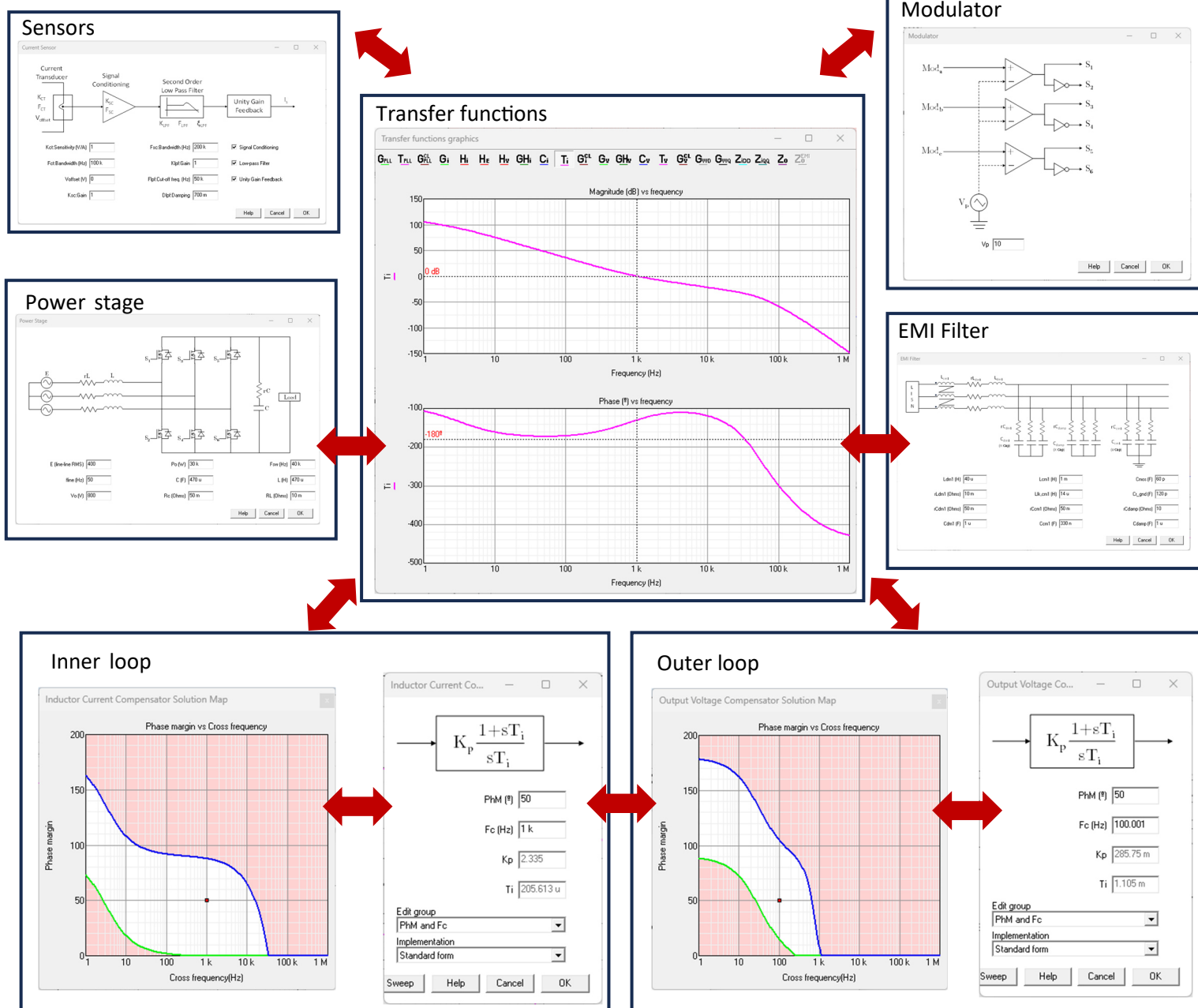




## Independent floating windows

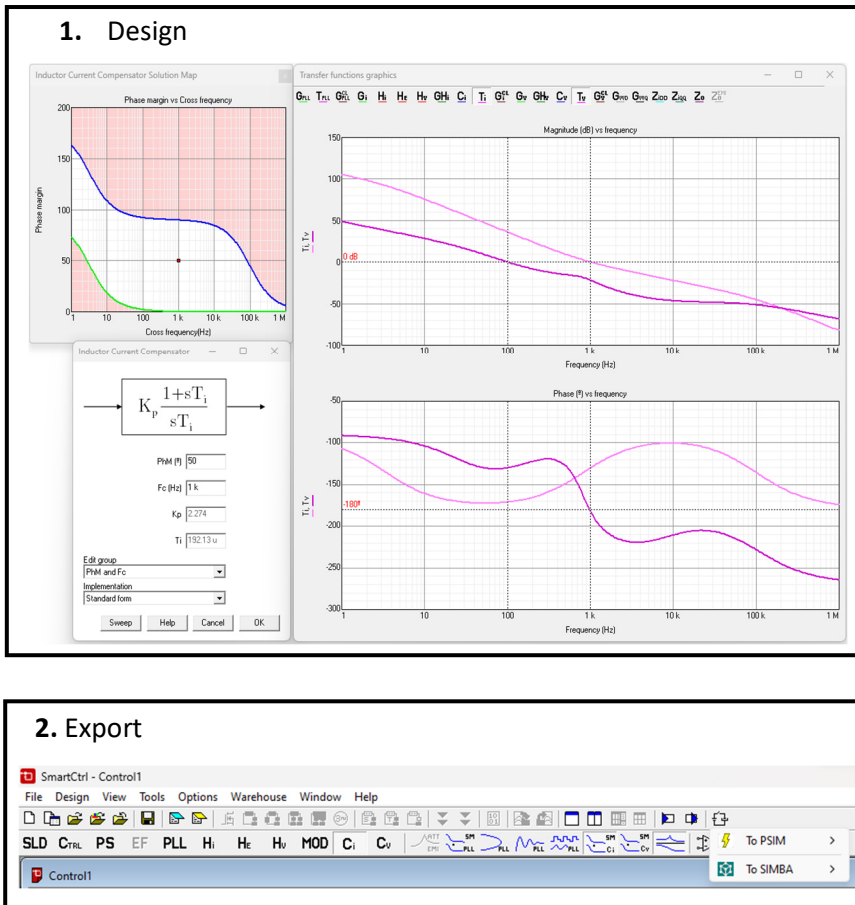
**SmartCtrl 2024.1** provides flexibility to customize the work panel, allowing users to select any required set of windows and position them freely within the work screen.

All windows are interconnected and exchange parameter values with each other in real time: value changes in one window immediately update the results in the rest. In addition, all windows are bidirectional: the same displayed parameters and graphs can be used as inputs (data collection) or outputs (results), depending on the case.

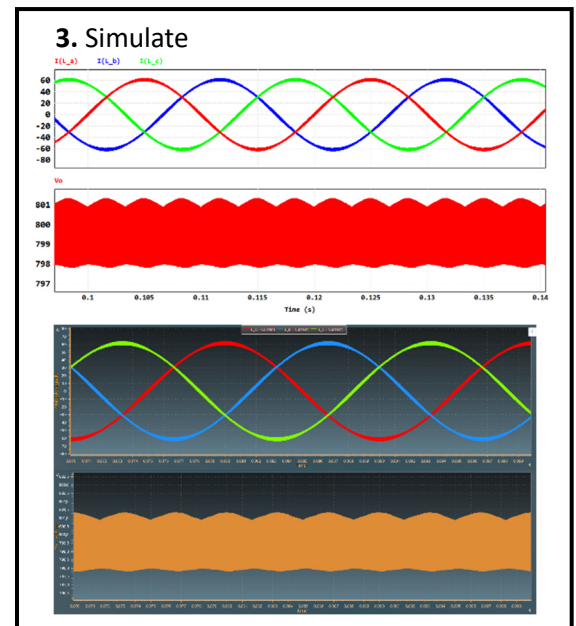




### Export the final design to third-party simulators

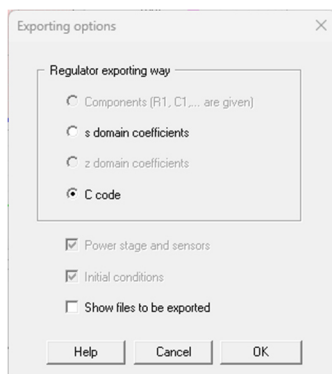


**SmartCtrl 2024.1** traditionally has provided a full seamless integration with PSIM, now that integration has been extended to SIMBA. This feature empowers users to efficiently validate their designs with speed and ease.



### Automated C-code generation

Additionally, the control design can now be exported in C code, which allows the user to have a starting template when implementing the control structure in a microcontroller or in a System on Chip SoC.



```
//Phase-Locked-Loop
//Theta calculation for the three-phase
teta_a = teta;
teta_b = teta - dos_pi_tercios;
teta_c = teta + dos_pi_tercios;

//Phase A
if (teta_a >= dos_pi){
    teta_a = teta_a - dos_pi;
} else {
    if (teta_a < 0){
        teta_a = teta_a + dos_pi;
    }
}

//Phase B
if (teta_b >= dos_pi){
    teta_b = teta_b - dos_pi;
} else {
    if (teta_b < 0){
        teta_b = teta_b + dos_pi;
    }
}

//Phase C
if (teta_c >= dos_pi){
    teta_c = teta_c - dos_pi;
} else {
    if (teta_c < 0){
        teta_c = teta_c + dos_pi;
    }
}

//Conversion from radians to index
index_a = teta_a*conv_rad_to_index;
index_b = teta_b*conv_rad_to_index;
index_c = teta_c*conv_rad_to_index;

//Ed, Eq calculation (Park's transformation)
ed_prev = Ea*full_cosine[index_a] + Eb*full_cosine[index_b] + Ec*full_cosine[index_c];
eq_prev = Ea*full_sine[index_a] + Eb*full_sine[index_b] + Ec*full_sine[index_c];

ed = (1/Gain_e)*(Kern)*(ed_prev);
eq = (1/Gain_e)*(-Kern)*(eq_prev);
```

