Triphase HPC Explained

Shaping Dynamic Response using Virtual Circuit Control™

Piet Vanassche, Triphase N.V.

Introduction

Controlling the currents and voltages in a lossless filter is a common power electronics control problem. Elementary cases concern the current through an inductor and the voltage over a capacitor. The left-hand panel in figure 1 illustrates a more complex setup whereby a switching inverter connects to the grid via a third-order ladder network. The filter stage, consisting of two inductors and a capacitor, has resonant poles that need to be controlled. On top of that, the output current needs to be as harmonic-free as possible. Similar problems are encountered with the realization of grid-emulating voltage sources using switching inverters and second or fourth-order filter sections.

Confronted with this problem, a circuit's designer's first impulse is most likely to try and introduce damping resistors in order to get rid of the filter resonances, probably modifying some of the filter components in the process. The result might be something as illustrated in the right-hand panel of figure 1. With the circuit being damped, it becomes much easier to control. Physical damping resistors however reduce the system's overall efficiency. Moreover, they constitute a thermal liability as large current levels may cause excessive heating.

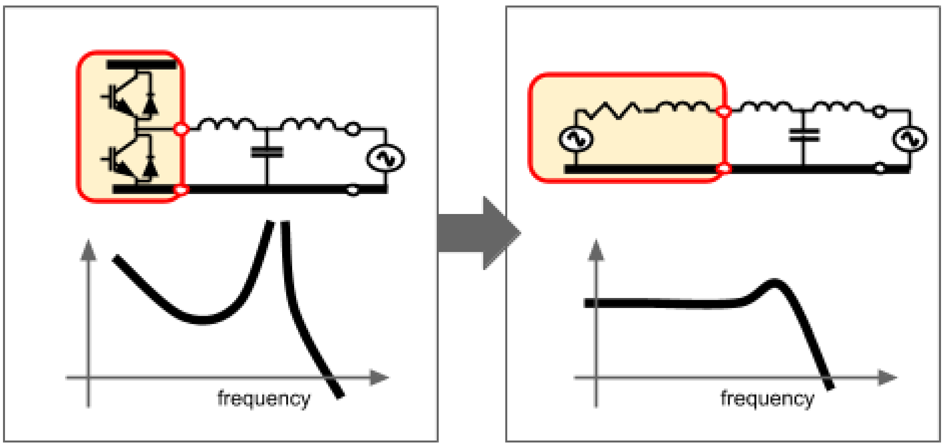


Figure 1: The left-hand side shows a switching inverter topology connecting to a grid via a lossless 3rd-order ladder network. Left uncontrolled, the filter exhibits resonant behaviour. In the right-hand side, this is remedied by proper resistive termination of the lossless ladder and an adjustment of the inverter-side inductance. The idea underlying Virtual Circuit Control™ is to emulate this behaviour in software.

Despite of the deficiencies, the simplicity of the approach above is charming. It is easy to understand both by circuit and control engineers. So why not try and emulate this concept in software! Comparing the left- and right-hand sides of figure 1, the idea is to program the inverter to emulate both the resistive termination as well as the adjustment of the inverter-side inductance. The combination of hardware components and software-emulated components is a virtual circuit with a properly damped low-pass characteristic. This approach is termed Virtual Circuit Control™.

This whitepaper explains Virtual Circuit Control™ in greater detail. It also introduces the Triphase High Performance Control (HPC) toolbox components that help you implement Virtual Circuits. The sections that follow respectively explain the role of the Virtual CIrcuit concept in the overall control hierarchy, how Virtual Circuit Control™ helps you shape dynamic behaviour and how to build more refined control strategies on top of your Virtual Circuit. All sections use current injection into an AC grid as a reference case.

Tackling one problem at a time

The problem of injecting current into an AC grid through a 3rd-order filter is depicted in figure 2. The current in the output inductor has to be controlled against a time-varying voltage to a specified harmonic pattern by proper switching of the inverter thereby avoiding excitation of filter resonances.This is a whole lot of problem to tackle all at once and a single-stage controller would be very abstract and difficult to configure. Triphase HPC therefore defines a multi-layer control hierarchy, each layer tackling a well-defined problem, i.e.

1. shaping the system dynamic behaviour and damping of the filter resonant poles
2. active harmonic control of the output current injected into the grid

By tackling these problems one at a time, the resulting control strategy becomes easier to understand and configure. Moreover, this hierarchical approach is readily extended, e.g. through active estimation of the grid impedance and adapting parameters in the underlying layers of control accordingly.

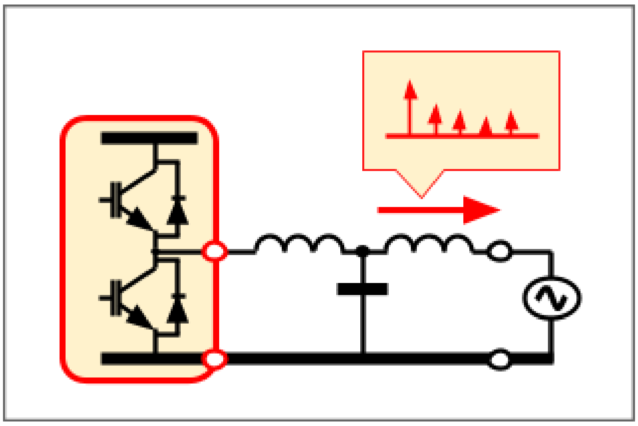


Figure 2: Inverter-driven current injection into the grid. The current in the output inductor has to be controlled against a time-varying voltage to a specified harmonic pattern by proper switching of the inverter thereby avoiding excitation of filter resonances.

Figure 3 details the layers in the control hierarchy facilitated by the Triphase HPC toolbox. The lowest level comprises the hardware and hardware interfaces. Next comes the Virtual Circuit Controller that shapes the system's dynamic behaviour. More refined control strategies, such as active harmonic control, build on top of this Virtual Circuit. The system is open for users to add their own, additional layers.

At hardware level, users interact directly with the raw current and voltage measurements as well as with the inputs of the inverter PWM generators. This gives complete low-level access to your Triphase setup. Of course, at this level, users have to deal with the nitty-gritty details ranging from proper measurement processing to taking care of resonances and all kinds of component non-idealities.

Virtual Circuit Control™ introduces an abstraction level that takes care of part of those nitty-gritty details. In particular, it handles filter resonances through the introduction of damping resistors and impedance adjustments[1]. As such, it defines the system's dynamic behaviour. Properties like step response times are defined at this level. With Virtual Circuit Control™ in place, users can deal with their system as if it was a terminated ladder network with a properly damped, low-pass frequency characteristic. By concept, this Virtual Circuit is much easier to handle.

Through Virtual Circuit Control™, the system presents itself as having a damped low-pass characteristic towards all overlying layers of control. For example, the active harmonic control uses this damped transfer function to compute the input voltage it has to apply to the virtual circuit in order to obtain an output current with the proper harmonic content.

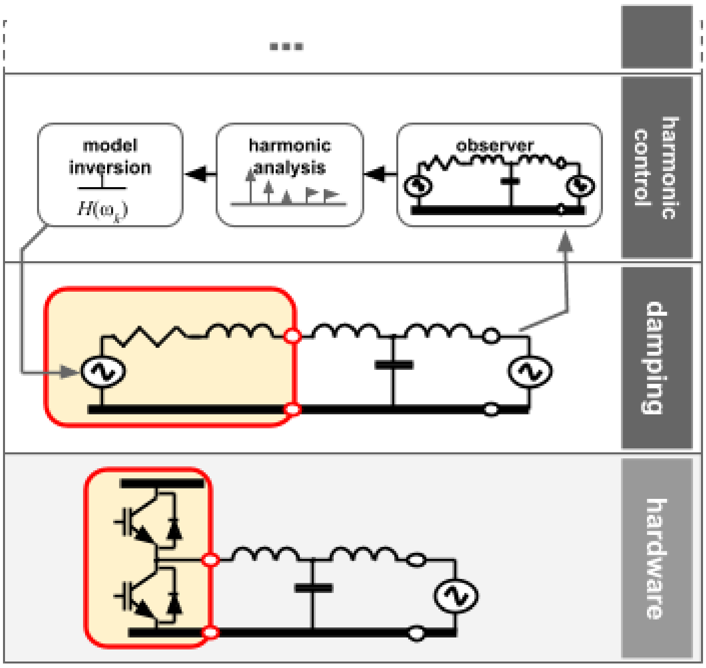


Figure 3: Triphase HPC control hierarchy. The lowest level comprises the hardware and hardware interfaces. Next comes the Virtual Circuit Controller that shapes the system's dynamic behaviour. More refined control strategies, such as active harmonic control, build on top of this Virtual Circuit.

Shaping dynamic behaviour through Virtual Circuit Control™

The basic idea underlying Virtual Circuit Control™ is depicted in figure 4. Proper switching of the inverter half bridges is used to emulate an inverter-side termination resistor, combined with an adjustment of the inverter-side inductance. Towards control layers higher up in the hierarchy, such as active harmonic damping, this combination of hard- and software components presents itself as a damped 3rd-order virtual circuit. This makes it much easier to reason upon for system and control engineers alike.

The Virtual Circuit concept is not limited to 3rd-order filters. It can actually be shown that it is applicable to ladder networks of any order [1]. Given a transfer function with a specific passband characteristic (maximally flat, equiripple, etc.) and specific locations of the transmission zeros, it is possible to show that there exists a resistor-terminated lossless ladder network, consisting of

[1] It actually also provides a basis for handling inverter and component non-idealities, but this will be the subject of another paper in the Triphase HPC Explained series.

inductors and capacitors, that will realize this transfer function[2]. Most of these components need to be realized in hardware. However, the inverter-side impedance can be software adjusted as done in figure 4 for a 3rd-order network. This allows a designer to select the physical inverter-side inductance in order to satisfy switching ripple requirements. Through software, the virtual overall inverter-side inductance can then either be increased or decreased compared to the actual physical value.

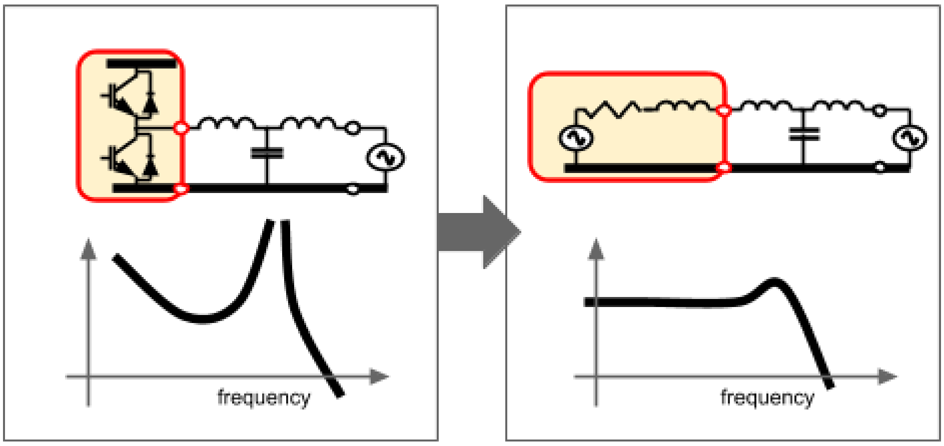
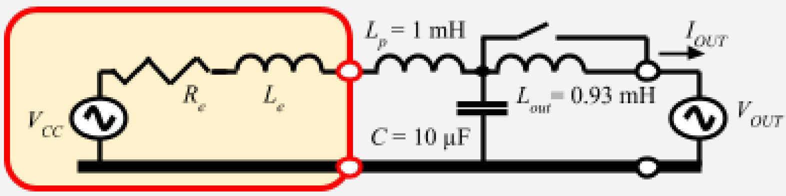


Figure 4: Virtual Circuit Control™ emulates an inverter-side termination resistor and inductance adjustment. Towards control layers higher up in the hierarchy this combination of hard- and software components presents itself as a properly damped virtual filter network that is much easier to manipulate.

Example – Shaping 2nd- and 3rd-order filter dynamics

The examples below build on the PM15 hardware components as listed in the figure below. As noted in the figure, the output-side inductance of any PMx filter stage can be bypassed, allowing the selection of either a 2nd- or 3rd-order output stage. The figure also marks the virtual components as emulated by Virtual Circuit Control™-driven inverter switching.



Second-order systems

When configured as a 2nd-order stage the resulting virtual circuit has a cut-off frequency and a damping factor given by

[2] These ladder network realizations have the additional advantage that they are quite robust with respect to parameter variations.

$$\omega_0 = \frac{1}{\sqrt{(L_p+L_e)C}}$$

$$\zeta = \frac{R_e}{2}\sqrt{\frac{C}{L_p+L_e}}$$

With two degrees available via R_e and L_e, cut-off frequency and damping factor can be tuned to any desired value. For example, without modification of the physical inductance, i.e. $L_e = 0$, the resonance frequency $f_0 = \omega_0/2\pi = 1592$ Hz. If we want to boost this to 2500 Hz, we need to set L_p+L_e = 405 µH or L_e = -595 µH . Hence, part of the physical inductance needs to get emulated away. The associated termination resistor R_e is then given by

$$R_e = 2\zeta\sqrt{\frac{L_p+L_e}{C}}$$

For critical damping at 2500 Hz, i.e. $\zeta = 1$, this yields R_e = 12.7 Ω.

Third-order systems

Third-order circuits are a bit more complicated. Here, there is no absolute freedom anymore in setting cut-off frequency and damping once the -physical- output inductance has been fixed. Without going into detail, we state following relations

$$\omega_0\tau_1 = 1$$

$$\omega_0 = \sqrt{\frac{1+2\zeta}{L_{out}C}}$$

$$L_e = \frac{L_{out}}{4\zeta^2+4\zeta} - L_p$$

$$R_e = \frac{1+2\zeta}{4\zeta^2+4\zeta}\omega_0 L_{out}$$

Here, ω_0 and ζ respectively represent the natural frequency and the damping factor of the transfer function's complex conjugate pair of poles. τ_1 is the time constant of the additional first-order component.

Again, we use the PM15 filter section for a numerical example. If we wish $\zeta = 0.8$, the equations above yield

$$f_0 = \frac{\omega_0}{2\pi} = 2661 \text{ Hz}$$

$$L_e = \text{-839 µH}$$

$$R_e = 7.02\ \Omega$$

Note the negative emulated inductance indicating that a large part of the physical inductance is emulated away. This makes sense in order to guide resonant currents towards the, also emulated, terminating resistor.

With the desired virtual circuit parameters fixed, it is up to the control software to properly emulate this behaviour. The fundamental control law to do so is straightforward. It simply requires combining the equation for the inverter current as determined by the physical and the virtual circuit, i.e.

$$L\frac{di_{inv}}{dt} = v_{inv} - v_C$$

$$(L_e + L_p)\frac{di_{inv}}{dt} = v_{CC} - v_C - R_e i_{inv}$$

Here, v_{CC} is the virtual circuit input voltage and v_C is the filter's capacitor voltage. Eliminating the time derivative of i_{inv} then yields

$$v_{inv} = \frac{L_p}{L_p+L_e}(v_{CC} - R_e i_{inv}) + \frac{L_e}{L_p+L_e} v_C$$

as the fundamental equation of the Virtual Circuit Controller.

Practical implementation of the control law derived above is a bit more complex than meets the eye. High-end control requires filter current and voltage estimations with proper compensation of controller sample delays as well as compensation of non-ideal behaviour caused by, amongst others, inverter dead-time and filter parameter variations. To this account, the Triphase HPC toolbox offers observer and control blocks taking care of this for 1st, 2nd and 3rd order filter stages. Observer details are described more fully in another Triphase HPC explained paper [2].

Current control with active harmonic compensation

By shaping the system dynamics as a properly damped low-pass filter, thereby eliminating filter resonances, the Virtual Circuit Controller removes an important sting from the overall problem. From this point on, control designers can interface with the virtual circuit rather than the hardware circuit itself. Hereby, the virtual circuit's damped low-pass characteristic is much easier to handle and reason upon than the original hardware circuit and its resonances.

Figure 5 illustrates a current control strategy build on top of the damped virtual circuit. It implements

$$V_{CC} = R_e I_{SP} + V_{out} + \Delta V_{CC}$$

as its control law. Here, V_{out} refers to the (right-hand) side output voltage, e.g. a 50 Hz grid, I_{out} is the associated output current while V_{CC} is the virtual circuit (left-hand side) input voltage.

The control law builds on the observation that, for low frequencies, i.e. frequencies much smaller than the virtual circuit's cut-off frequency, the virtual circuit is dominated by the -virtual- resistor R_e . Therefore, the first two terms express the input voltage needed to control the desired current through this resistor. These terms correspond with the lower part of the block diagram in figure 5.

The remaining term ΔV_{CC} is computed to actively counteract undesired output current harmonics. The error current is translated to an equivalent correction voltage by inverting the transfer function of the virtual circuit. In order to avoid stability problems due to right-hand plane zeros, the inversion is performed in the frequency domain. Correction voltage levels are integrated per harmonic with the time constant T_N determining the speed of compensation.

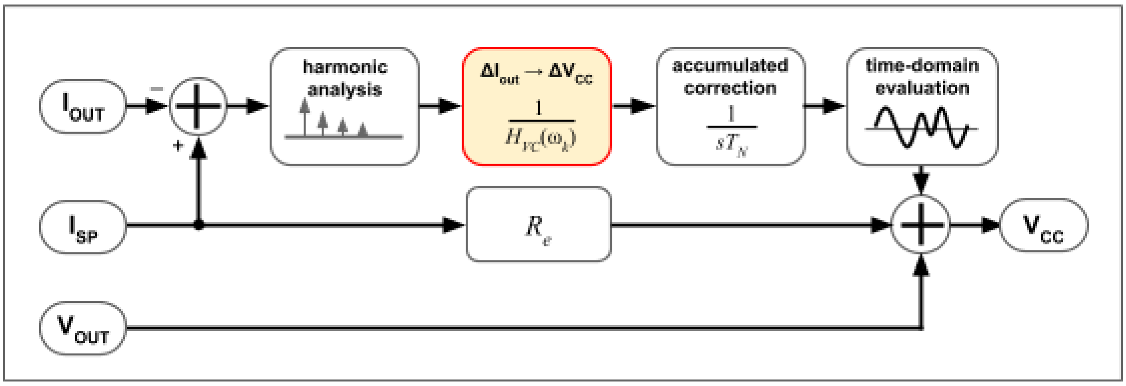


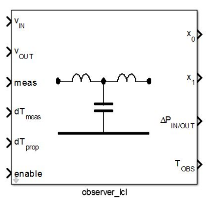
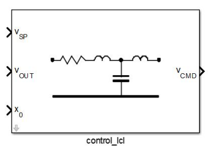
Figure 5: The current controller with active harmonic compensation computes inputs for the damped virtual circuit. It inverts the characteristic of the virtual circuit in order to compute setpoint corrections in the harmonic domain. In this figure, V_{out} refers to the (right-hand) side output voltage, e.g. a 50 Hz grid, I_{out} is the associated output current while V_{CC} is the virtual circuit (left-hand side) input voltage.

The Virtual Circuit Control™ principle makes algorithms like the one in figure 5 straightforward to implement. The underlying reason for this is the decoupled control of transient and steady-state behaviour. With the original filter circuit first being stabilized, it becomes much easier to devise controllers that optimize steady-state behaviour. In the above, we discussed current control through a 3rd-order output filter. Of course, similar reasonings hold for filter topologies of all orders.

Virtual Circuit Control™ in Triphase HPC

The Triphase HPC toolbox foresees a number of blocks to help users realise their own virtual circuits. These blocks are listed in the table below. Blocks always come in pairs. For a given filter topology, there is an observer block that estimates the filter's state as well as offset errors onto input and output excitations. Observer blocks are also able to compensate for sample delays by looking ahead into the future. A second, controller, block implements the sizing and emulation of the actual virtual circuit.

More details on the use and configuration of these blocks is provided in the accompanying Triphase HPC step-by-step tutorial on Virtual Circuit Control™ [4].

	**observer_lcl:** Kalman observer for a third-order LCL filter network. The observer estimates the filter's internal current and voltages as well as offset errors on input and output voltage. The observer is also able to compensate sample delays by predicting future behaviour. Works in conjunction with the *control_lcl* block.
	**control_lcl:** Virtual Circuit Controller implementation for a terminated third-order filter network. This block computes the proper virtual component values and evaluates the virtual circuit's control law. Works in conjunction with the *observer_lcl* block.

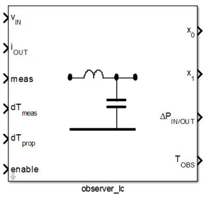
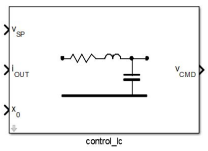
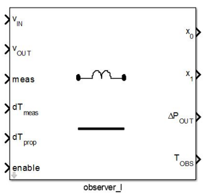
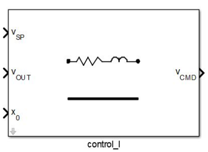
	**observer_lc:** Kalman observer for a second-order LCL filter network. The observer estimates the filter's internal current and voltages as well as offset errors on input voltage and output current. The observer is also able to compensate sample delays by predicting future behaviour. Works in conjunction with the *control_lc* block.
	**control_lc:** Virtual Circuit Controller implementation for a terminated second-order filter network. This block computes the proper virtual component values and evaluates the virtual circuit's control law. Works in conjunction with the *observer_lc* block.
	**observer_l:** Kalman observer for a first-order inductor-only filter. The observer estimates the inductor current current as well as offset errors on input and output voltage. The observer is also able to compensate sample delays by predicting future behaviour. Works in conjunction with the *control_l* block.
	**control_l:** Virtual Circuit Controller implementation for a terminated third-order filter network. This block computes the proper virtual component values and evaluates the virtual circuit's control law. Works in conjunction with the *observer_lcl* block.

Table 1: Overview of virtual circuit-related Triphase HPC blocks

References

[1] A.J. Casson and E. Rodriguez-Villegas, *A Review and Modern Approach to LC Ladder Synthesis*, Journal of Low Power Electronics and Applications (http://www.mdpi.com/journal/jlpea), pp. 20-44, 2011

[2] Triphase HPC Explained white paper*, Under the hood of Triphase Virtual Circuit Control*, Triphase N.V., 2014