Digital control of switching mode power supplies

Simone Buso

University of Padova – ITALY

Dept. of Information Engineering – DEI

Lesson 2

Basic organization of the digital controller

Signal conditioning and sampling
Synchronization between sampling and PWM
Quantization noise, arithmetic noise, limit cycle oscillations (LCOs)
Example of basic digital current control implementation

Study of limit cycle oscillations in MPPT algorithms

Gradient or hill-climbing MPPT algorithms (Perturb and Observe)
Limit cycles oscillations in P&O MPPT algorithms
Mathematical model of hill-climbing MPPT algorithms
Method to predict LCO occurrence and amplification
Optimal design of the MPPT algorithm

Digital proportional integral controller

Approximated dynamic model of delay effects

Digital current mode control: basic organization

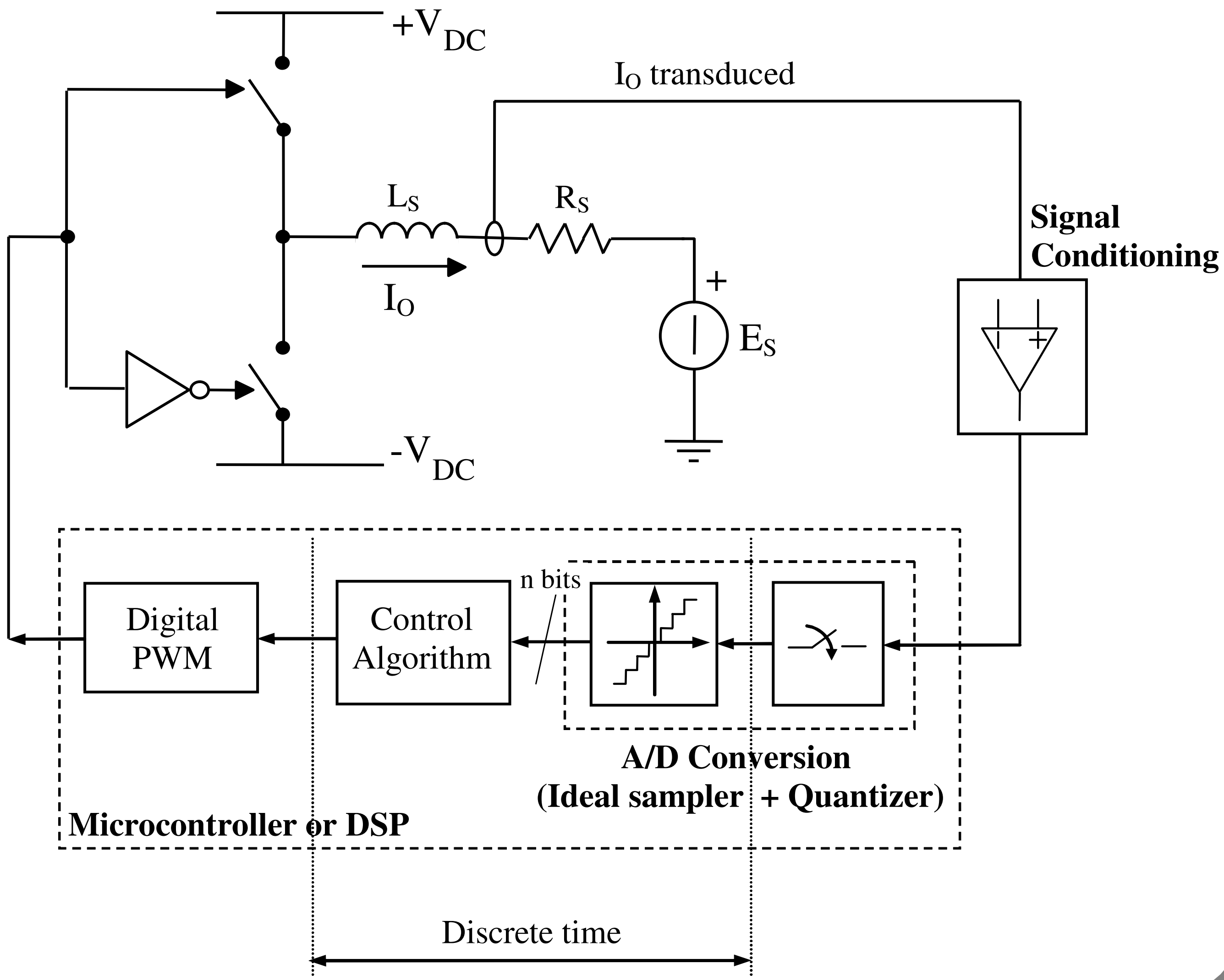
The data acquisition path

We assume the digital controller is developed using a microcontroller (mC) or digital signal processor (DSP) unit, with suitable built-in peripherals. Almost every mC and several low cost DSP units, typically identified as motion control DSPs or industrial application DSPs, include the peripheral circuits required by our set-up.

These are basically represented by an *analog to digital converter* (ADC) and a *PWM unit*. The *data acquisition path* for our current controller is very simple, being represented by the cascade connection of a current sensor, a properly designed signal conditioning electronic circuit and the ADC.

The *conditioning* circuit has to guarantee that: *i)* the sensor signal is amplified so as to fully exploit the input voltage range of the ADC, *ii)* the signal is filtered so as to avoid *aliasing* effects.

Digital current mode control: basic organization



Digital current mode control: basic organization

ADC voltage range exploitation

The full exploitation of the ADC input voltage range is a key factor to reduce the *quantization effects* that may undermine control stability and/or reduce the quality of the regulation. The number N_e of effective bits, that are used for the internal representation of the input signal samples is given by:

$$N_e = n - \text{floor}\left(\frac{\log_{10} \frac{FSR}{S_{PP}}}{\log_{10} 2} \right)$$

where S_{PP} is the peak to peak amplitude (in Volts) of the transduced input signal, *FSR* is the ADC full scale range (in Volts) and *n* is the ADC bit number.

Digital current mode control: basic organization

ADC voltage range exploitation

A little complication we typically find when designing the conditioning circuit is related to the *sign* of the input signal.

It is quite common for the transduced current signal to be bipolar (i.e. to have both positive and negative sign), while the lower bound of the ADC voltage range is almost always zero. To take care of that, the conditioning circuit has to offset the input signal by a half of the ADC FSR.

Given the expected peak to peak amplitude of the VSI output current and considering a suitable safety margin for the detection of over-current conditions, due to load transients or faults, it is immediately possible to determine the gain required to exploit the ADC full scale range.

Digital current mode control: basic organization

Aliasing phenomena

The *aliasing* phenomenon is a consequence of the violation of Shannon's theorem, which defines the limitations for the exact reconstruction of a uniformly sampled signal.

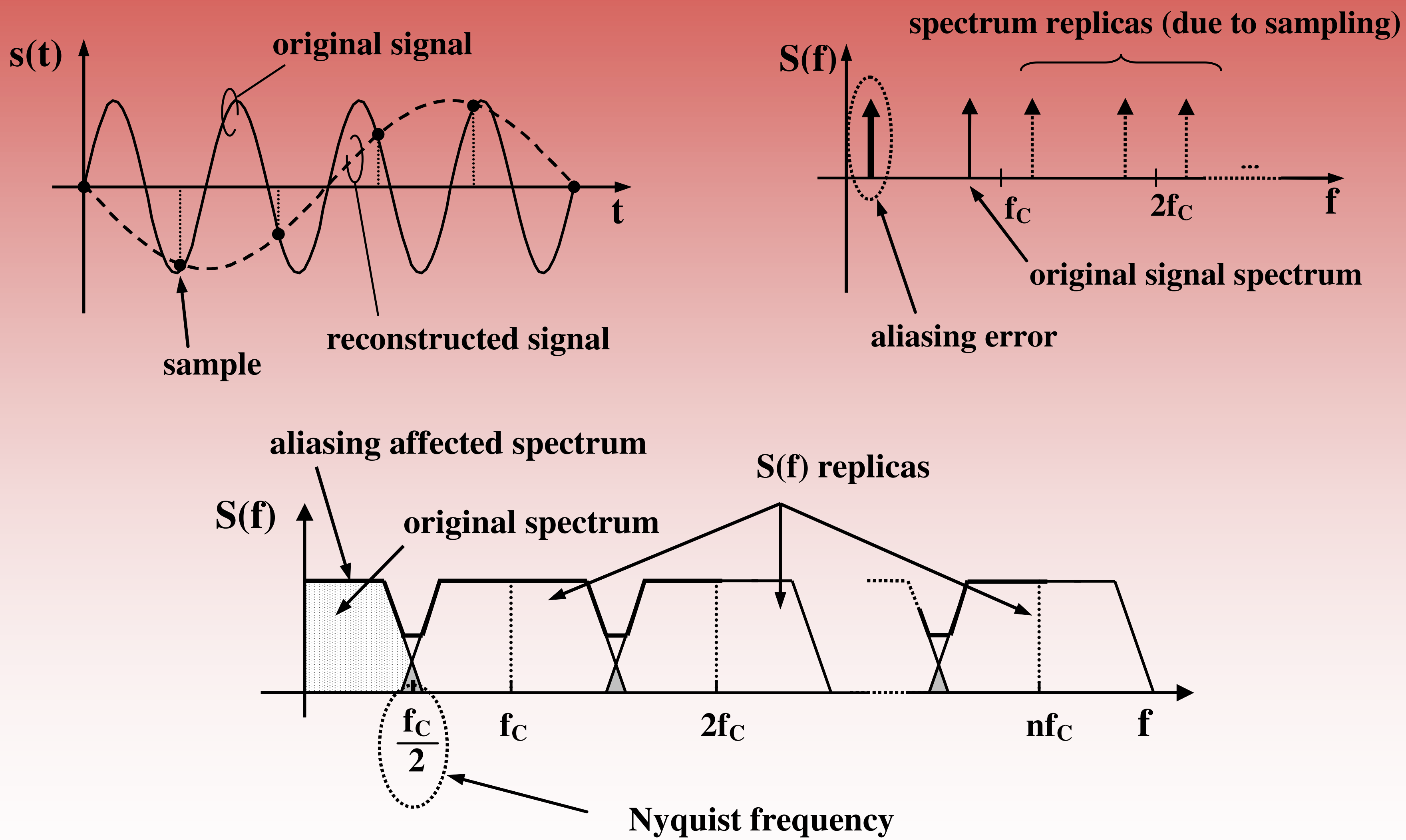
The theorem shows that there is an upper bound for the sampled signal bandwidth, beyond which perfect reconstruction, even by means of ideal interpolation filters, becomes impossible and aliasing phenomena appear.

The limit frequency is called the Nyquist frequency and is proved to be equal to a half of the sampling frequency, f_C. In general, we will have to limit the frequency spectrum of the sampled signal by filtering, so as to make it negligible above the Nyquist frequency.

This condition will determine the bandwidth and roll-off of the filter synthesized by the conditioning amplifier.

Digital current mode control: basic organization

Aliasing phenomena



Digital current mode control: basic organization

ADC equivalent model

In mathematical terms, the analog to digital conversion process can be modelled as the cascade connection of an *ideal sampler* and a *n* bit uniform *quantizer*.

The former is defined as a sampler whose output is a stream of *null duration* pulses, each having an amplitude equal to that of the input signal at the sampling instant (Dirac pulses).

Its function is to model the actual sampling process, i.e. the transformation of the time variable from the continuous domain to the discrete domain, where time only exists as integer multiples of a fundamental unit, the sampling period.

Digital current mode control: basic organization

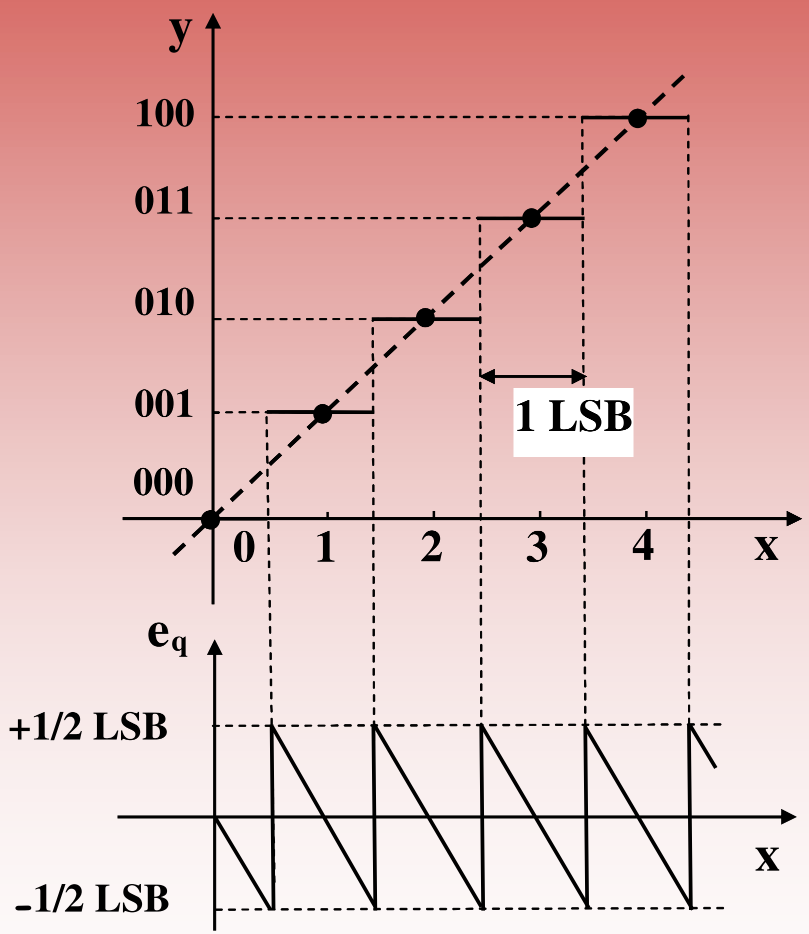
ADC equivalent model

The latter is taken into account to model the *loss of information* implied by what can be interpreted as a *coding* procedure, where a continuous amplitude signal, i.e. a signal whose instantaneous level can vary with continuity in a given range of values, is transformed into a discrete amplitude signal, i.e. a digital signal, whose instantaneous level can only assume a finite number of values in the same given range.

Because the possible discrete values can be interpreted as integer multiples of a fundamental unit, the quantization step *Q,* or, equivalently, the *least significant bit* LSB, the quantizer is called “uniform”.

Digital current mode control: basic organization

ADC equivalent model



Typical trans-characteristic diagram for a uniform quantizer.

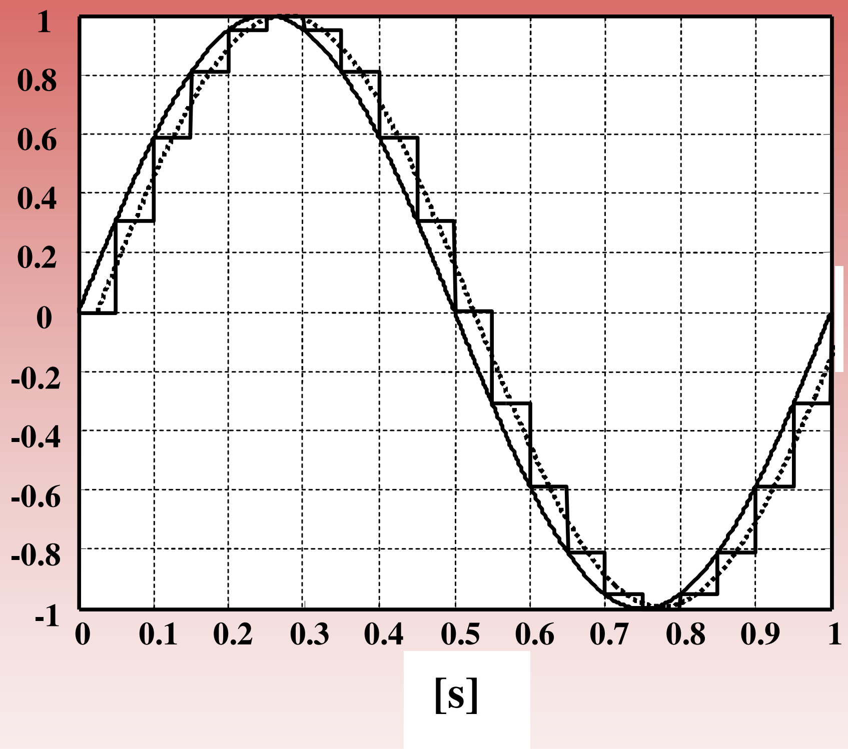
Quantization noise e_q is added to the signal as a result of analog to digital conversion.

This can be interpreted as the loss of some of the information associated to the input signal, inherent to the analog to digital conversion and unavoidable.

As far as the dynamic behaviour of the ADC is concerned, it should be evident that both the quantizer and the ideal sampler are essentially instantaneous functions, that do not contribute to the dynamics of the system.

Digital current mode control: basic organization

Interpolation process: how to return to continuous time



Zero Order Hold (ZOH) approximation of the interpolation process.

Neglecting the presence of the control algorithm, we can describe this model simply considering that, in order to reconstruct the continuous time signal from the discrete time input samples, each sample value is held constant for the entire duration of the sampling period.

We recognize immediately that the process implies a *delay effect,* equivalent to half a sampling period.

Digital current mode control: basic organization

Synchronization of PWM and sampling processes

In order not to violate the Shannon's theorem, the sampling process should proceed at a very high frequency, so high that the spectrum of the sampled signal might be considered negligible at the Nyquist frequency, even if a significant ripple is observable.

This requires a sampling frequency at least one order of magnitude higher than the switching frequency.

Unfortunately, hardware limitations do not allow the sampling frequency to become too high: our controller implementation is based on standard microcontroller or DSP hardware.

In the typical case, the sampling frequency will be set equal either to the switching frequency, or, if this is consistent with the available digital PWM implementation, to two times the switching frequency.

Digital current mode control: basic organization

Synchronization of PWM and sampling processes

But, if this is what we do, the Shannon's theorem conditions will always be violated!

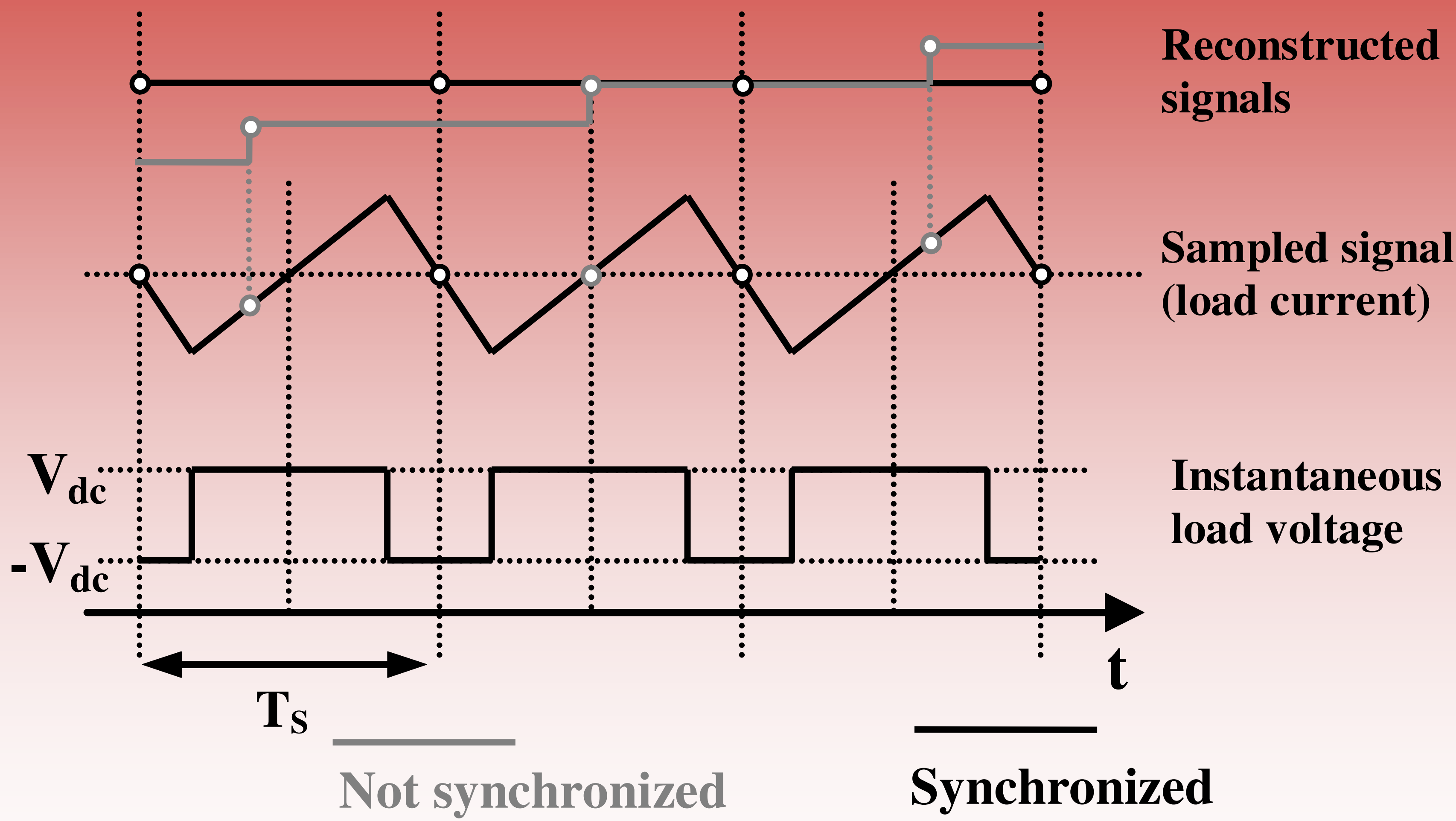
This is one of the key issues in digital control applications to power electronic circuits: the typically recommended high ratio between sampling frequency and sampled signal bandwidth will never be possible.

However, this condition, rather than detrimental, is normally advantageous for the controller effectiveness. The reason for this lies in *synchronization.*

If the sampling and switching processes are suitably synchronized, the effect of aliasing is the automatic reconstruction of the average value of the sampled signal, which is *exactly* what has to be controlled.

Digital current mode control: basic organization

Synchronization of PWM and sampling processes



Digital current mode control: basic organization

Synchronization of PWM and sampling processes

Synchronization allows the reconstruction of the average signal value any time the sampling takes place in the middle of the switch on period or in the middle of the switch off period (or both, if double update mode is possible).

Instead, if the switching and sampling frequencies are different, low frequency aliasing components will be created in the reconstructed signal.

Even if the sampling and switching frequencies are set equal, there still can be a zero frequency error in the reconstruction of the average sampled signal, in case the sampling instants are not coincident with the beginning and/or the half of the modulation period.

This is generally a minor problem, since the current regulator will often be driven by an external loop that, typically including an integral action, will compensate for any steady state (or very low frequency) error in the current trajectory.

Digital current mode control: basic organization

Quantization and arithmetic noise

The typical implementation of analog to digital conversion in microcontrollers and DSPs associates a *binary* code to the amplitude values of the sampled signal. In the case of the uniform quantizer, the rule to associate a binary code *N* to any given signal sample *x* is very simple, and can be mathematically expressed as:

$$\begin{cases} \left(N - \frac{1}{2}\right) \cdot Q < x < \left(N + \frac{1}{2}\right) \cdot Q \quad \Rightarrow \quad x_q = N \\ Q = \frac{FSR}{2^n} = LSB \end{cases}$$

where *n* represents the ADC bit number and, as was previously described, if FSR represents the full scale range, in Volts, of the ADC, then *Q* is the ADC quantization step, equal to one least significant bit (LSB)

Digital current mode control: basic organization

Quantization and arithmetic noise

Q represents the *minimum variation of input signal* x that *always* causes the variation of at least one bit in the binary code associated to x_q, the coded signal.

Therefore, any variation of signal x smaller than Q is not always able to determine some effect on x_q.

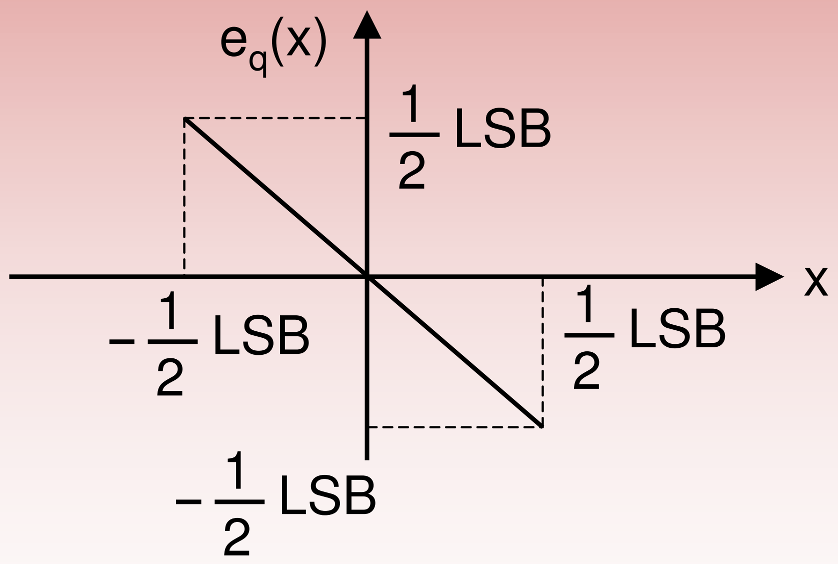
This simple observation shows us that the quantization process actually implies the loss of some of the information associated to the original signal x.

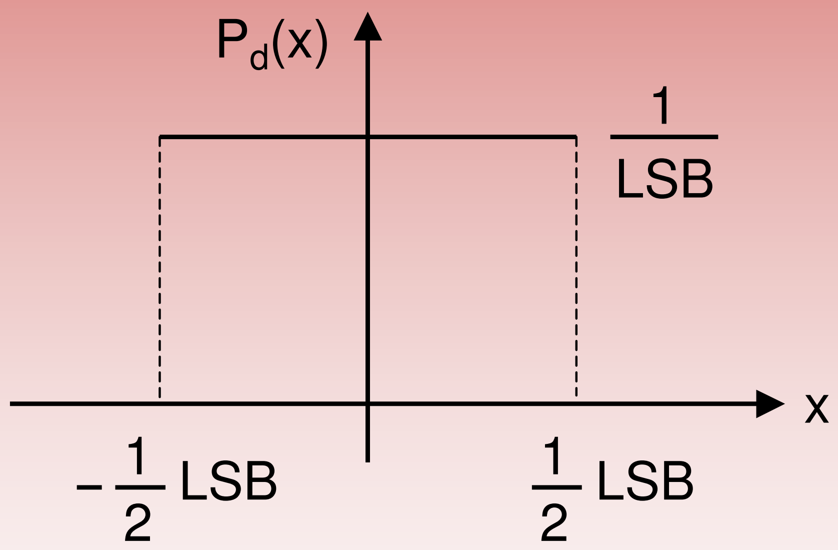
It is a common approach to model this effect as an additive noise, superimposed to the signal.

Digital current mode control: basic organization

Quantization and arithmetic noise

In order to simplify the mathematical characterization of the quantization noise, the stochastic process associated to it, $e_q(x)$, is assumed to be not correlated to signal x and to present uniform in probability density $P_d(x)$:





Digital current mode control: basic organization

Quantization and arithmetic noise

Under these assumptions, the statistical power of $e_q(x)$ can be found to be given by:

$$\sigma_q^2 = \frac{1}{LSB} \int_{-\frac{1}{2}LSB}^{+\frac{1}{2}LSB} e_q^2(x) \, dx = \frac{LSB^2}{12}$$

Digital current mode control: basic organization

Quantization and arithmetic noise

It is then possible to derive a very useful relation that expresses the *maximum* signal to noise ratio (SNR) of an ADC, as a function of its number of bits. This turns out to be given by:

$$\mathrm{SNR} = 10 \cdot \log_{10}\left(\frac{12}{8} \cdot 2^{2n}\right) = 6.02 \cdot n + 1.76\ [\mathrm{dB}]$$

Digital current mode control: basic organization

Quantization and arithmetic noise

Bit number n	LSB value [%] with respect to the FSR	Theoretical SNR [dB]
8	0.39	49.9
10	0.1	62
12	0.02	74
14	0.006	86

Resolution and theoretical signal to noise ratio (SNR) of an AD converter as functions of the bit number n.

Digital current mode control: basic organization

Quantization and arithmetic noise

There are at least two other major forms of quantization that always take place in the implementation of a digital control algorithm: *i) arithmetic quantization* and *ii) output quantization*.

The former is nothing but an effect of the finite precision that characterizes the arithmetic and logic unit (ALU) used to compute the control algorithm.

The finite precision determines the need for truncation (or rounding) of the controller coefficients' binary representations, so as to fit them to the number of bits available to the programmer for variables and constants.

In addition, it may determine the need for truncation (or rounding) after multiplications. In general, the effect of coefficient and multiplication result truncation (or rounding) is a distortion of the controller's frequency response, i.e. the shift of the system poles, that can have some impact on the achievable performance.

Digital current mode control: basic organization

Quantization and arithmetic noise

Both truncation and rounding effects can be modelled as another type of quantization and so as an equivalent noise, of arithmetic nature, added to the signal.

Predicting the amplification of arithmetic noise within a closed loop control algorithm by pencil and paper calculations is a really tough job.

To check the control algorithm operation to this level of detail, the only viable option is its complete, low level simulation, based on a model that includes the emulation of the adopted controller arithmetic unit.

Severe arithmetic quantization problems are seldom encountered, being confined to extremely demanding applications or to applications where the use of 8 bit microcontrollers is the only viable option and the emulation of a higher precision arithmetic is not possible for memory or timing constraints.

Digital current mode control: basic organization

Quantization and arithmetic noise

Output quantization is related to the truncation (or rounding) operation inherent in the digital to analog conversion that brings the control algorithm output variable back from the digital to the continuous time domain.

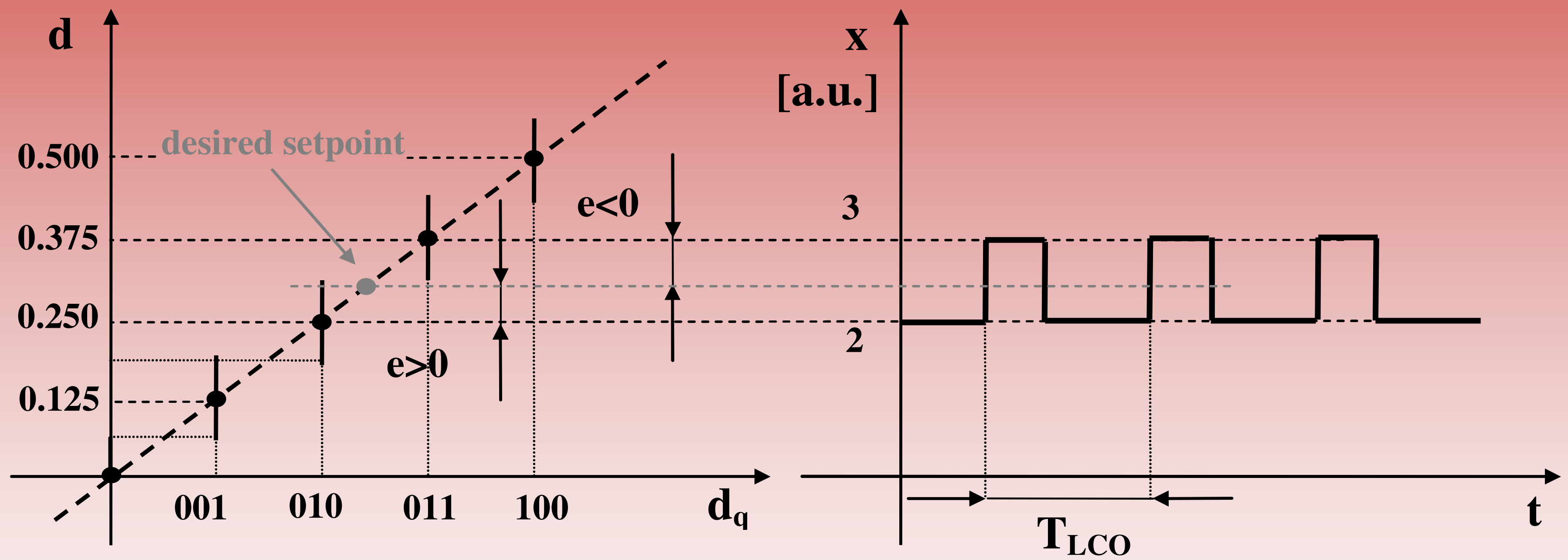
In our case, this function is actually inherent in the digital PWM process. The reduction of the control variable output (in our case the desired duty-cycle) bit number, as is needed in order to adapt it to the PWM duty-cycle register, represents a quantization noise source.

Remember that, unless a very high clock to modulation frequency ratio is available, the effective number of bits that might be used to represent the duty-cycle is always much smaller than the typical variable bit number (16 or 32).

The most unpleasing effect of output quantization may be the occurrence of a peculiar type of instability, specific of digital control loops, that is known as *Limit Cycle Oscillation, LCO.*

Digital current mode control: basic organization

Limit Cycle Oscillations - LCOs



Example of limit cycle oscillation

Digital current mode control: basic organization

Limit Cycle Oscillations - LCOs

Variable *d* is the duty-cycle of our switching converter, whose “desired” set-point is the particular value we need to apply to bring the converter to the steady-state.

Variable *x* may be associated, for example, to the converter average output current. Unfortunately, the desired set-point for *d* is not anyone of the possible outputs, because of output quantization.

As a result, we will either apply a bigger than needed duty-cycle, causing the current increase beyond the steady state level, or a lower than needed duty-cycle, causing the current decrease below the steady state value.

Indeed, the converter output current can be considered proportional to the integral of the inverter average output voltage, that is in turn, proportional to the duty-cycle. Commutations between the two states are determined by the current controller, that reacts to the current error build-up by changing the duty-cycle

Digital current mode control: basic organization

Limit Cycle Oscillations - LCOs

It is possible to formulate two *necessary,* but *not sufficient,* conditions to prevent limit cycle occurrence.

The *first one* is to ensure that the variation of one DPWM level, i.e. 1 LSB of the duty-cycle digital representation, here denoted as q_{DPWM}, does not determine a variation of the controlled output variable *x(t)*, in steady-state conditions, greater that the quantization level of *x(t)*, here denoted as q_{ADC}.

Thus, if we define as *G(s)* the transfer function between the duty-cycle, *d,* and the controlled variable, *x(t*), the first necessary condition for the elimination of LCOs is:

$$\boxed{q_{DPWM} G_{dc} < q_{ADC}}$$

Digital current mode control: basic organization

Limit Cycle Oscillations - LCOs

The *second* condition is the presence of an integral action in the controller. This can be explained considering that, if only a proportional term (or a proportional-derivative term) is included in the adopted controller, a minimum quantized error on the controlled variable $x(t)$ determines a variation on the average converter voltage equal to $G_{dc} \cdot K_P \cdot q_{ADC}$ (even considering the quantization of the DPWM to be infinite).

Since $G_{dc} \cdot K_P$ is usually much greater than one, this variation is much greater than q_{ADC} and, consequently, the first condition is not satisfied. Therefore, in order to comply with it, a lower amplification of the minimum quantized error on the input variable must be ensured.

This always happens when an integral action is included in the control algorithm.

Digital current mode control: basic organization

Limit Cycle Oscillations - LCOs

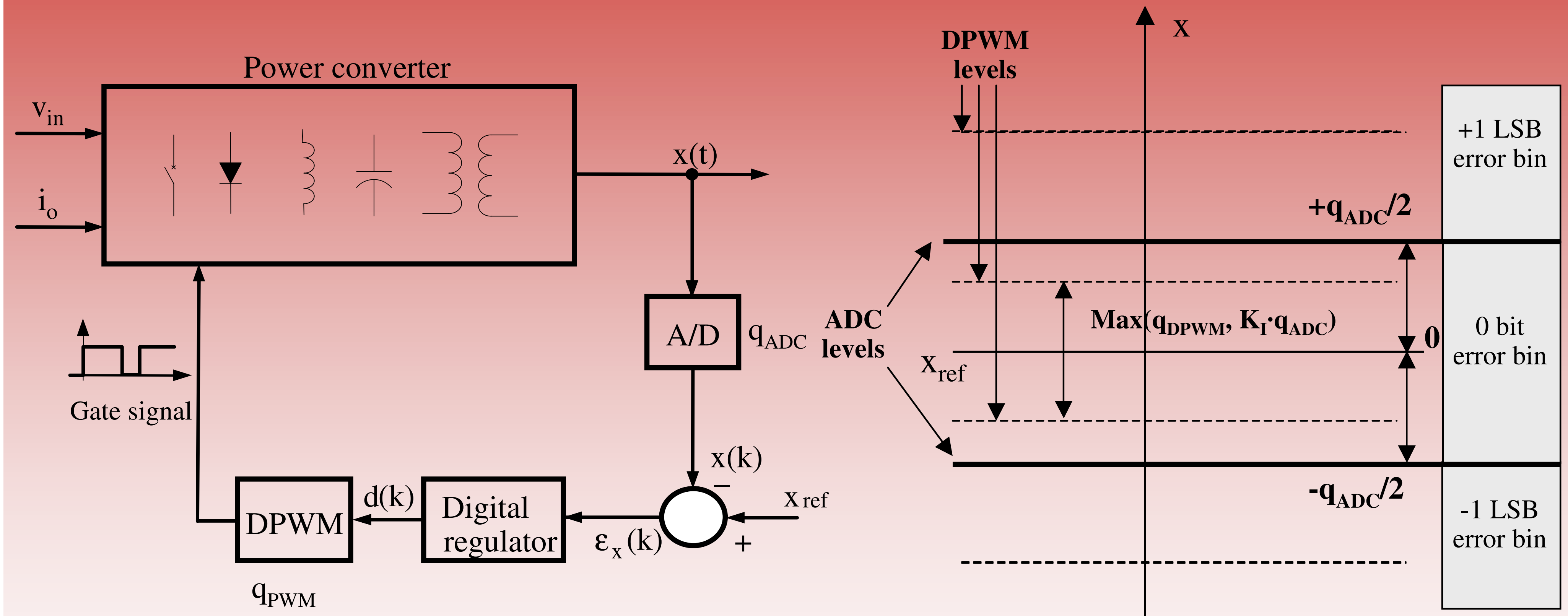
In that case, the integral gain *induces* a smaller quantization effect on the DPWM, since the minimum variation of the duty-cycle, due to the minimum quantized error on *x(t)*, is now equal to $K_I \cdot q_{ADC}$, with K_I normally much smaller than K_P. In addition to that, the following condition has to be satisfied:

$$\boxed{K_I \, G_{dc} < 1}$$

which actually imposes an upper limit to K_I. This guarantees that the first condition is satisfied. The simultaneous verification of both conditions, where we can now define the DPWM resolution as the *maximum* between its *physical,* hardware quantization and what we have called the *induced* quantization, determined by the integral term, is necessary to make the elimination of LCOs theoretically possible

Digital current mode control: basic organization

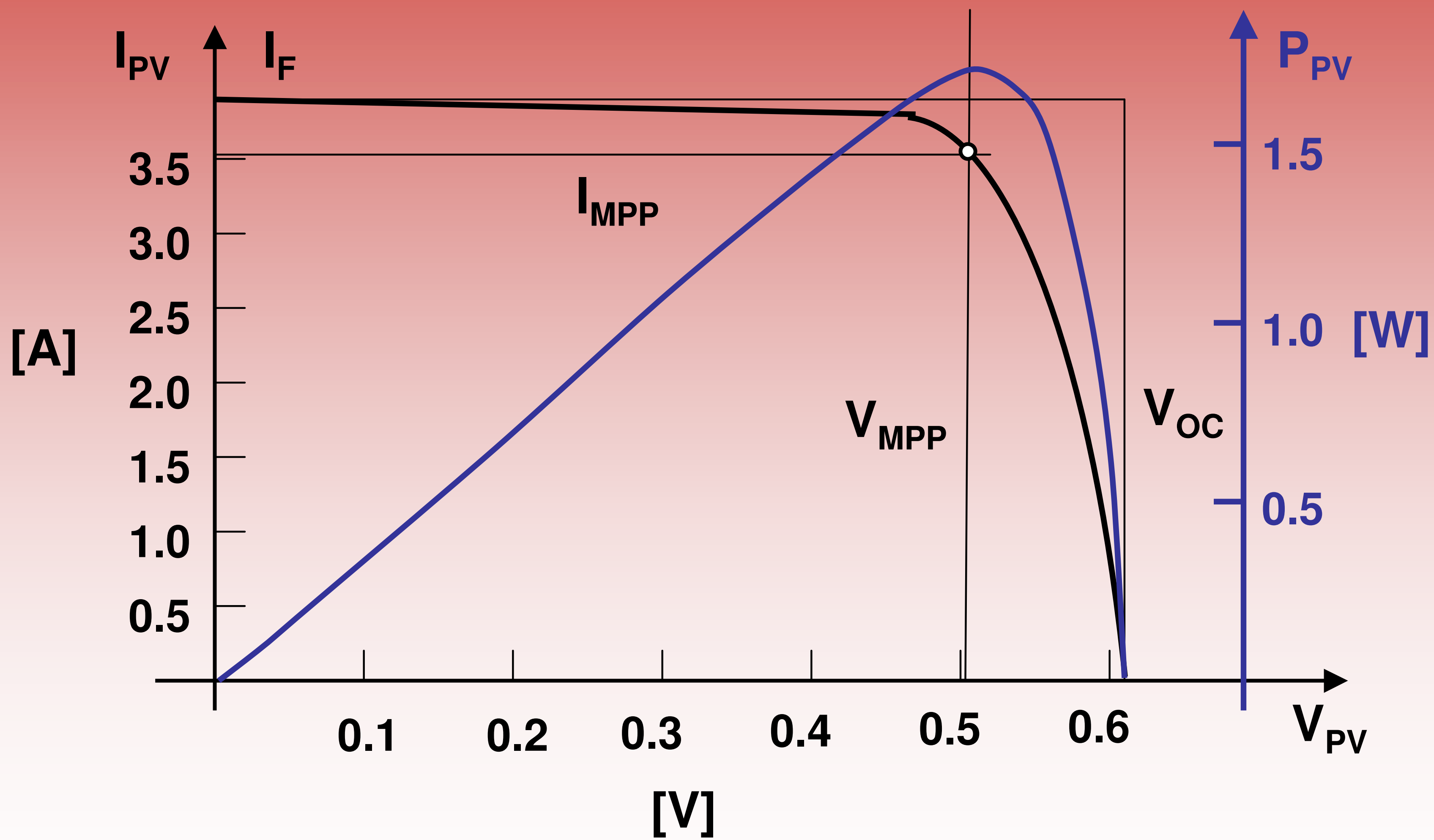
Limit Cycle Oscillations - LCOs



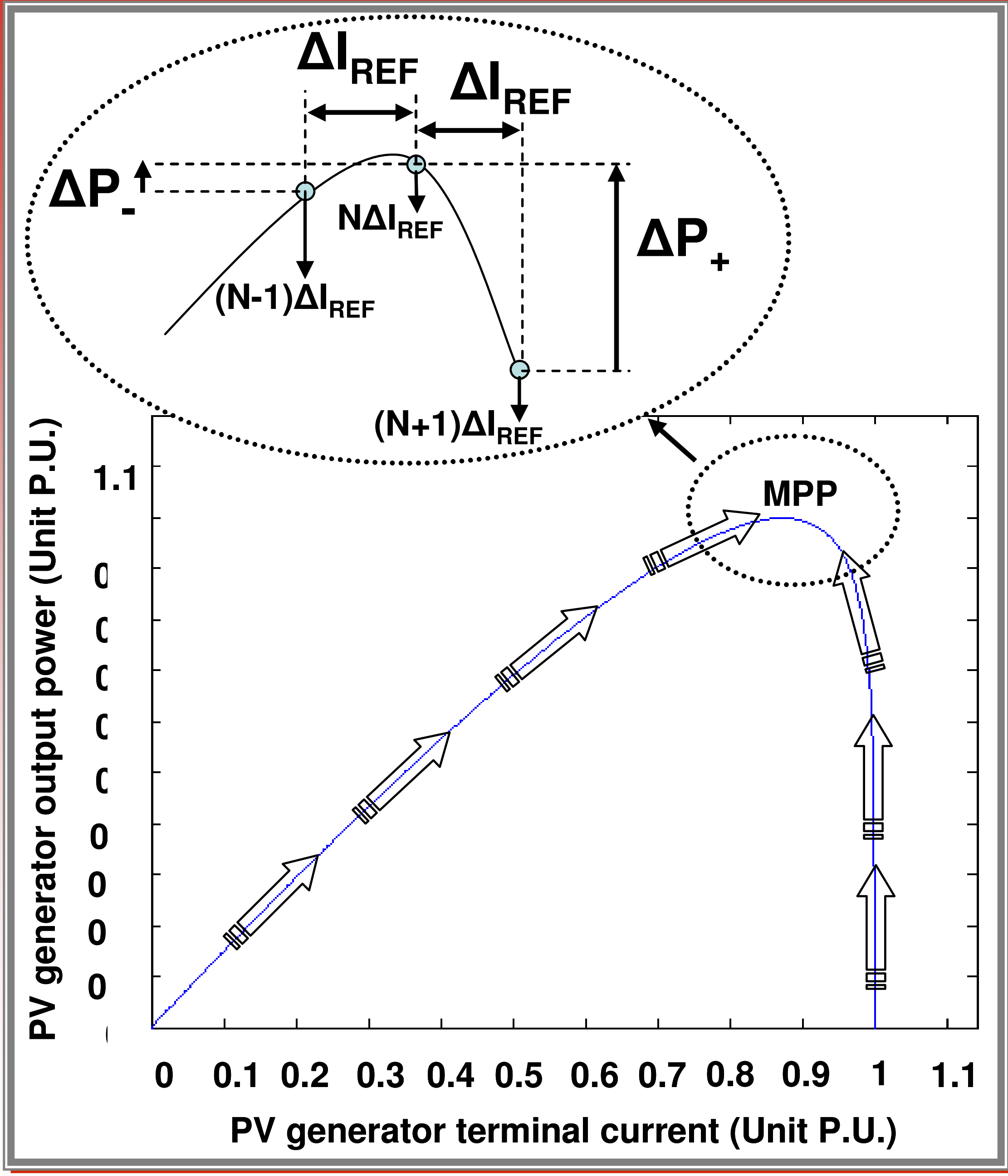
Conditions to prevent limit cycle oscillations in a closed loop digital regulation.

Study of limit cycle oscillations (LCOs) in MPPT algorithms

Perturb and Observe (P&O) maximum power point tracking



MPPT algorithms



Any MPPT algorithm is aimed at achieving stable operation of the PV generator on its maximum power point.

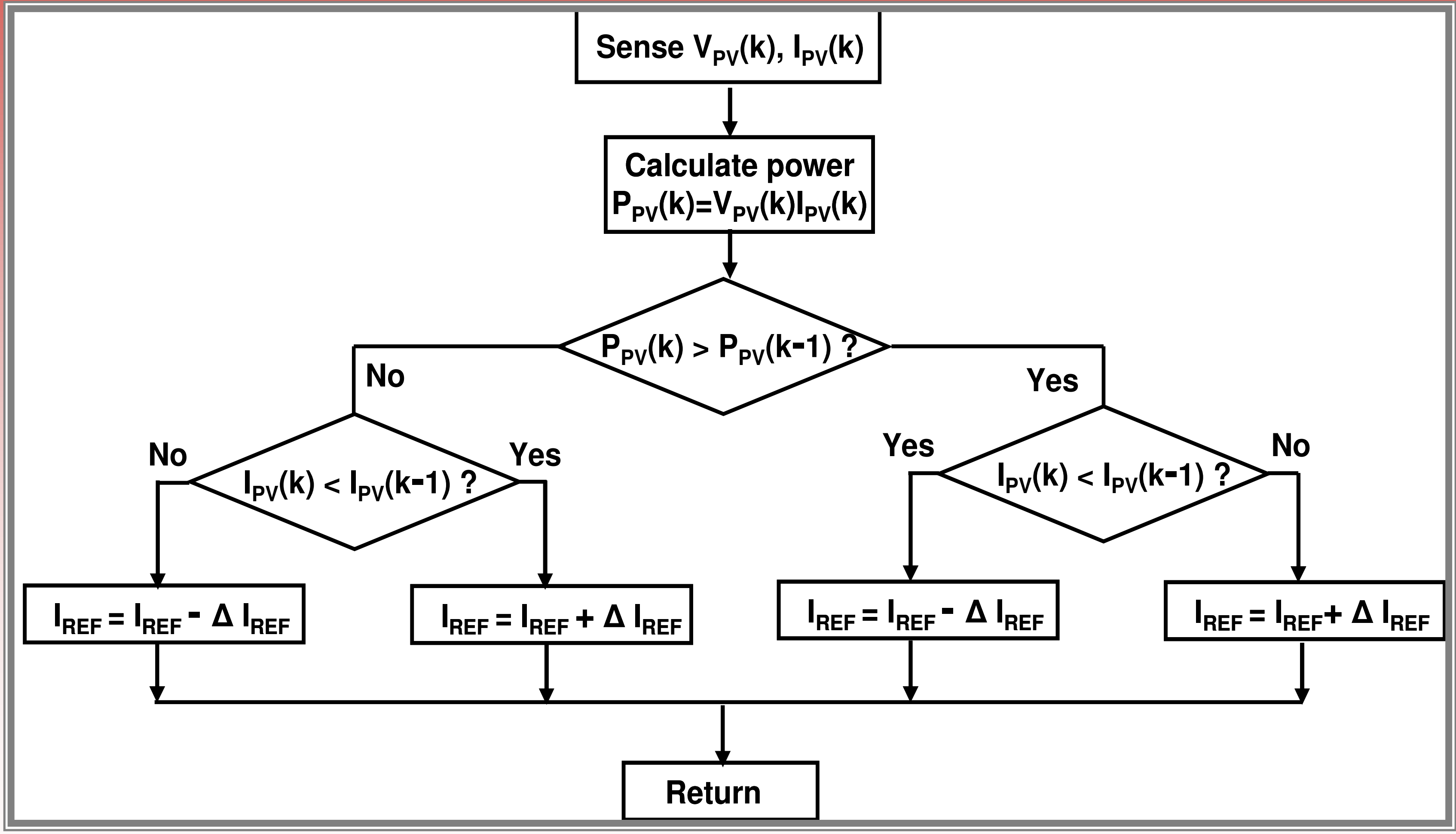
In P&O algorithms, periodic perturbations of the generator operating point are applied by the *interface converter.*

The variations of the extracted power are measured at each step and used to drive the generator towards the MPP.

Unfortunately, all algorithms, like the P&O, i.e. based on power derivative sign, imply the occurrence of limit cycles.

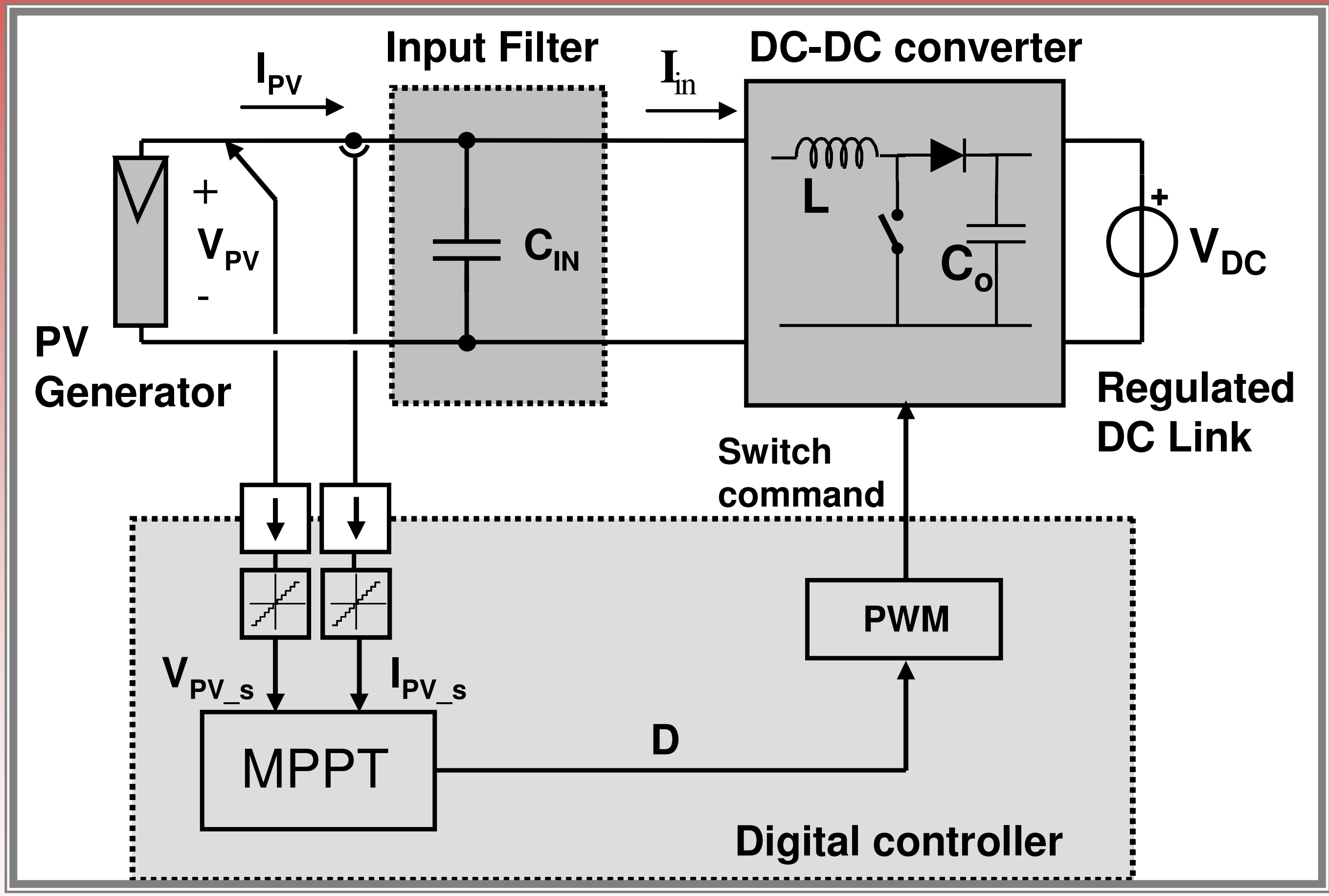
Study of limit cycle oscillations (LCOs) in MPPT algorithms

Perturb and Observe (P&O) maximum power point tracking



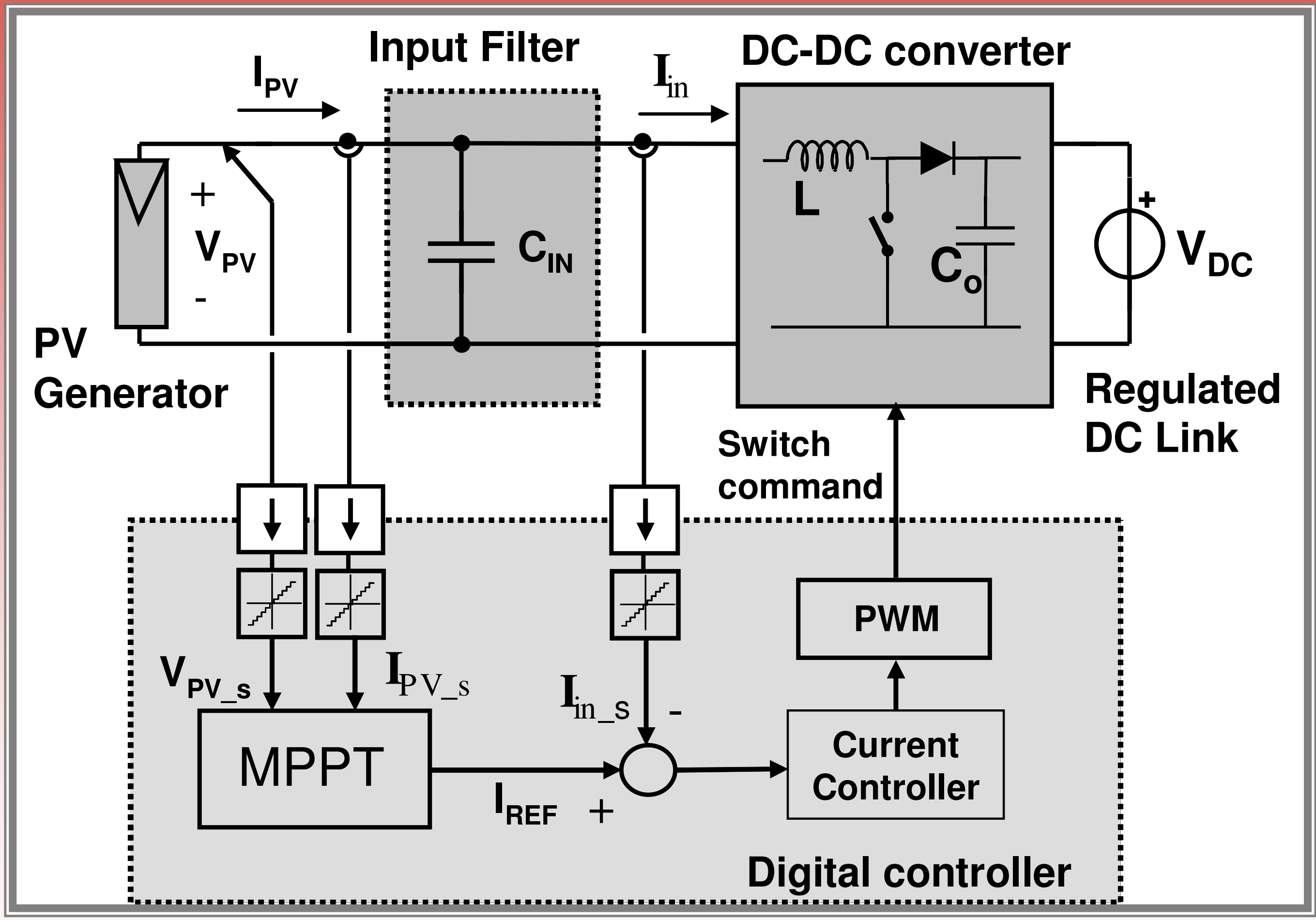
Study of limit cycle oscillations (LCOs) in MPPT algorithms

Perturb and Observe (P&O) maximum power point tracking



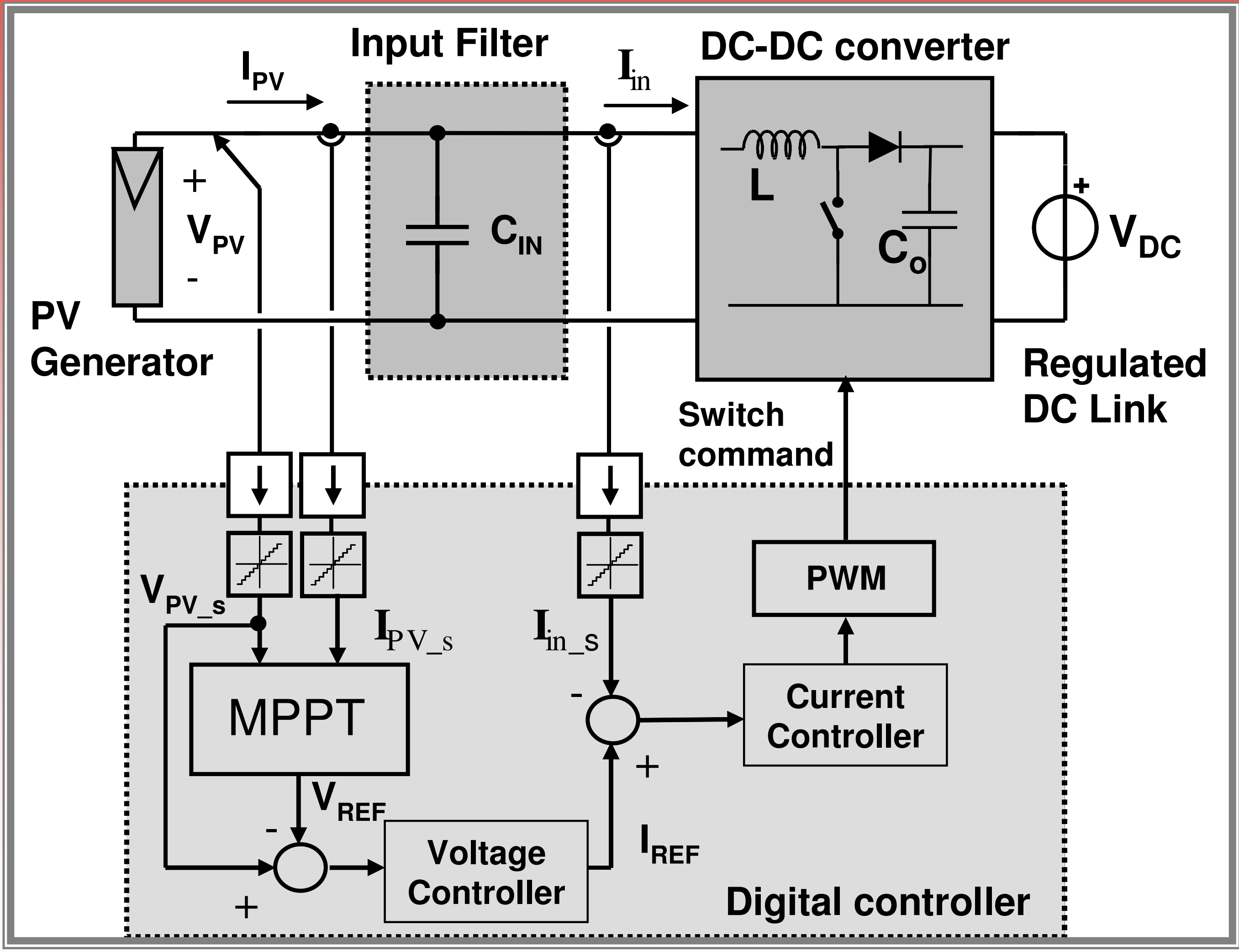
Study of limit cycle oscillations (LCOs) in MPPT algorithms

Perturb and Observe (P&O) maximum power point tracking

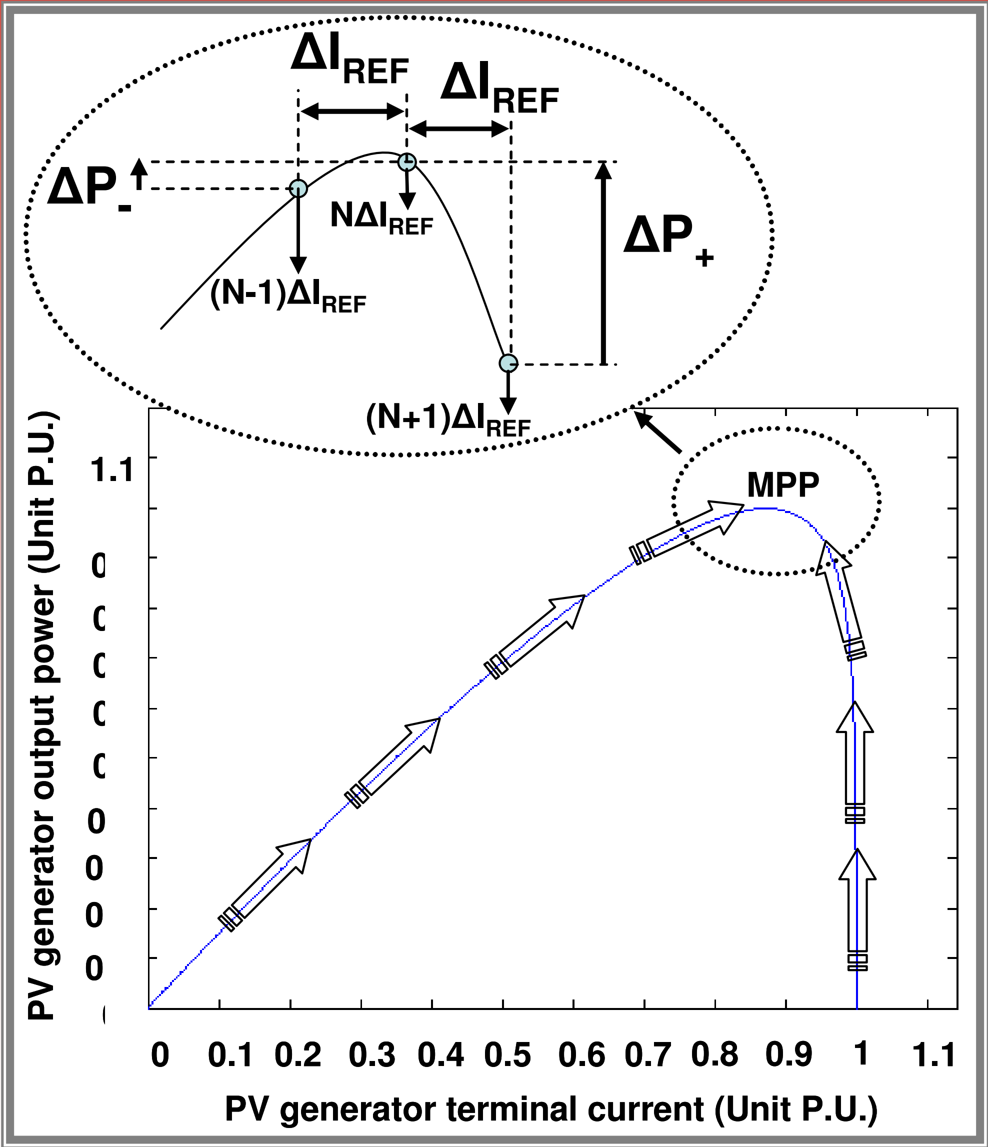


Study of limit cycle oscillations (LCOs) in MPPT algorithms

Perturb and Observe (P&O) maximum power point tracking

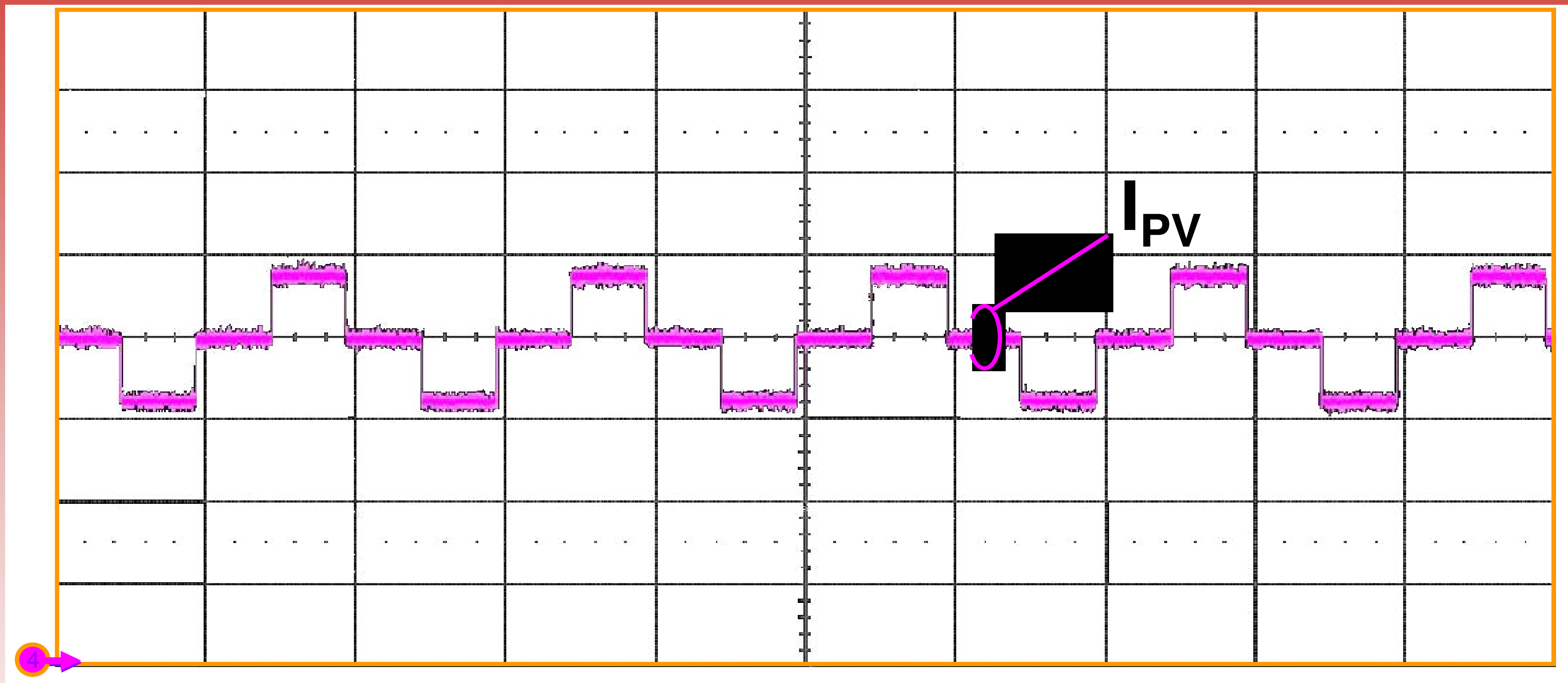


Study of limit cycle oscillations (LCOs) in MPPT algorithms



The current reference value providing the power level nearest to the maximum power point is $N \times \Delta I_{REF}$, being N a suitable integer. Changing I_{REF} from $(N-1) \times \Delta I_{REF}$ to $N \times \Delta I_{REF}$, the PV generator supplied power increases of a quantity ΔP_{-}. If this variation is measurable by the algorithm, i.e. if the input quantization is fine enough that the numerical representation of this quantity differs from zero, at the next algorithm's iteration, the current reference will be increased to $(N+1) \times \Delta I_{REF}$. Then, a power decrease equal to ΔP_{+} is determined and the current reference is set back to $N \times \Delta I_{REF}$. As this implies a power increase, at the following step, I_{REF} will be set again to $(N-1) \times \Delta I_{REF}$, leading to a new decrease in the extracted power. Clearly, a persistent oscillation between $(N-1) \times \Delta I_{REF}$ and $(N+1) \times \Delta I_{REF}$ is established.

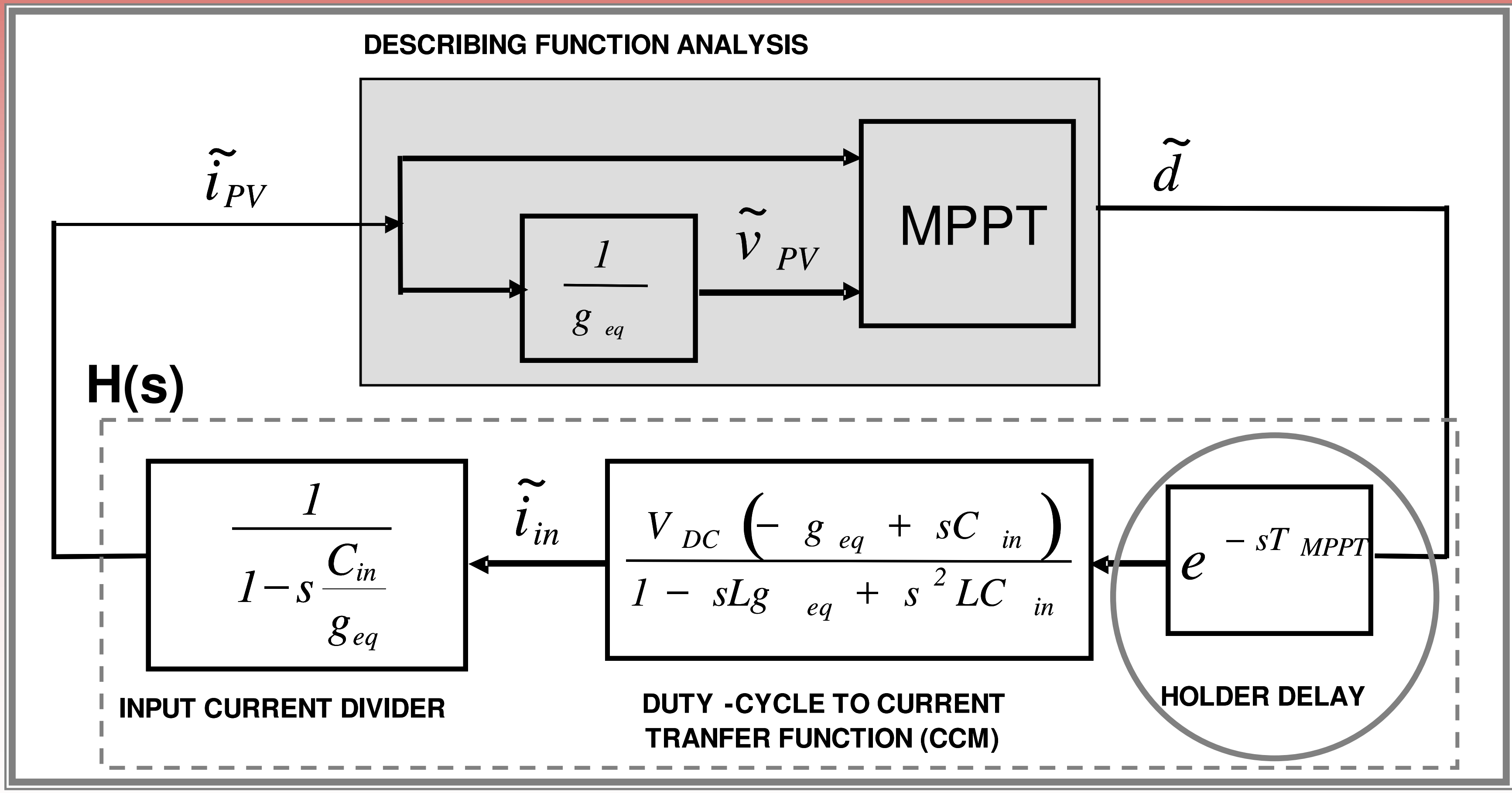
Study of limit cycle oscillations (LCOs) in MPPT algorithms



Measured limit cycle oscillation of the P&O MPPT at F_{MPPT} = 200 Hz: the plot represents the PV generator current, I_{PV} (100 mA/div, 10 ms/div). The ΔI_{REF} value amounts to 12 input LSBs.

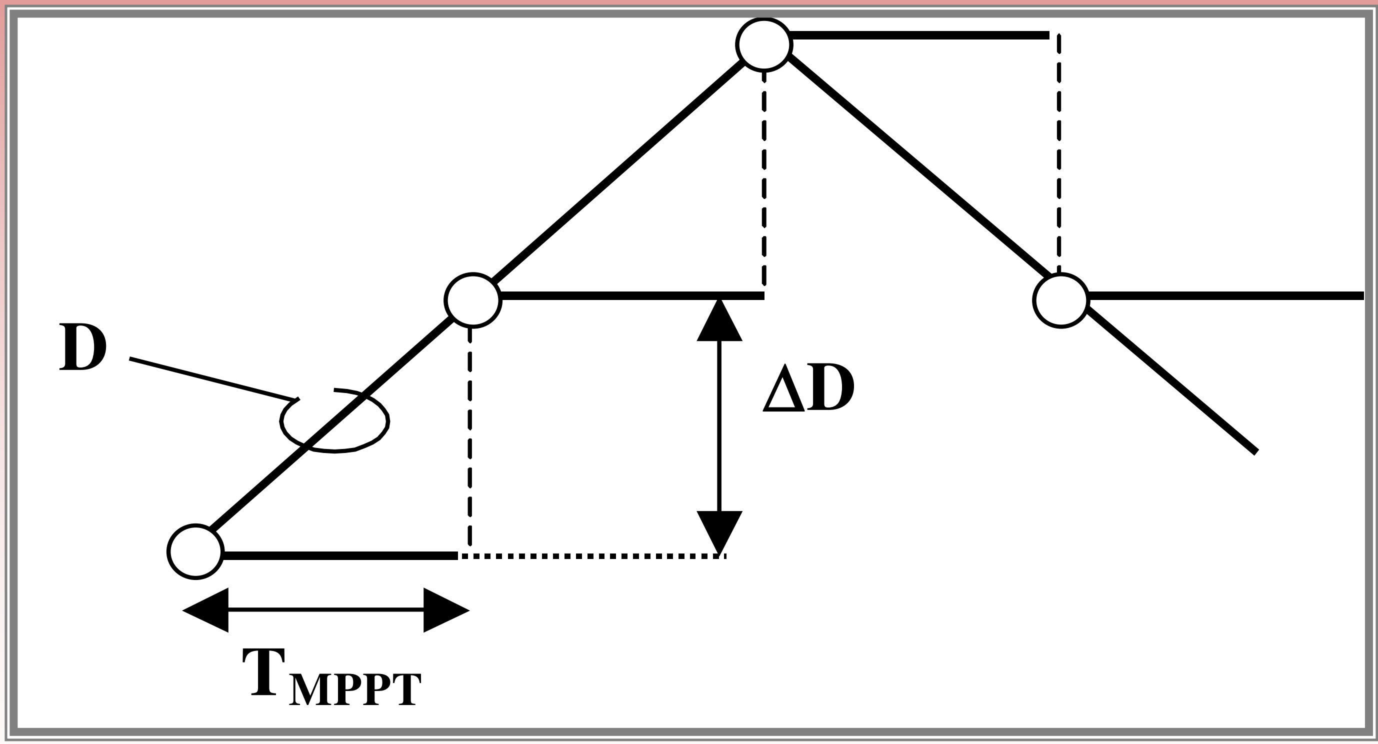
Study of limit cycle oscillations (LCOs) in MPPT algorithms

It is possible to define a small signal model for each of the above organizations. Considering the case of a boost converter with direct duty-cycle control, the model is built approximating the digital controller with an *equivalent continuous time* system. This can be formally done by a zero order hold interpolation of the original discrete time system.



Study of limit cycle oscillations (LCOs) in MPPT algorithms

Assuming the holder delay to be equal to the MPPT algorithm iteration period, $T_{MPPT} = 1/F_{MPPT}$, and observing that the current controller's sampling frequency is normally much higher than the MPPT algorithm's one, the current controller can be modeled as an equivalent continuous time regulator, with sufficient accuracy, at least for frequencies significantly lower than the current loop's Nyquist limit.



Study of limit cycle oscillations (LCOs) in MPPT algorithms

The PV generator has been modeled by its equivalent conductance at the MPP, defined as

$$\boxed{g_{eq} = \left.\frac{dI_{PV}}{dV_{PV}}\right|_{V_{PV}=V_{MPP}}}$$

Due to the PV generator I-V characteristic, the equivalent conductance is always negative. The incremental conductance has been used to determine the small signal relation between converter input current perturbation, $\tilde{i}_{in}$, and PV generator current perturbation, $\tilde{i}_{PV}$.

The current control loop has been represented as a simple first order low pass transfer function, with $\tau_I = 1/\omega_I$ and $\omega_I = 2\pi \cdot BW_I$ where BW_I is the current loop bandwidth.

Study of limit cycle oscillations (LCOs) in MPPT algorithms

Describing function of P&O MPPT

Modeling the MPPT algorithm in terms of a rational transfer function is impossible, due to its non linear nature. However, we can model the non linear component of the feedback loop by means of the so-called *describing function* method.

In order to apply the method, we first need to derive a continuous time equivalent of the algorithm, such as:

$$\boxed{D(t) = \frac{\Delta D}{T_{MPPT}} \cdot \int_0^t sign(\dot{P}_{PV} \cdot \dot{I}_{PV})\, d\tau + D(0)}$$

Study of limit cycle oscillations (LCOs) in MPPT algorithms

Describing function of P&O MPPT

Given the low pass characteristics of the current loop and generator input current divider (Fig. 6), the study can be carried out assuming that, if a persistent oscillation is established, the time variant part of the MPPT block input signal is approximately sinusoidal.

$$\begin{cases} I_{PV} = \bar{I}_{PV} + \tilde{i}_{PV} \cong \bar{I}_{PV} + i_{PV}\, sin(\omega t) \\ V_{PV} = \bar{V}_{PV} + \tilde{v}_{PV} \cong \bar{V}_{PV} + v_{PV}\, sin(\omega t) \end{cases}$$

where the steady state (dc) and small signal (approximately sinusoidal) components of the generator current and voltage are explicitly indicated and $\omega = 1/T$ is the oscillation angular frequency. The PV generator instantaneous power is given by the following expression:

$$P_{PV}(t) = \bar{V}_{PV}\bar{I}_{PV} + \bar{I}_{PV} v_{PV} \cdot sin(\omega t) + \bar{V}_{PV} i_{PV} \cdot sin(\omega t) + \ldots$$
$$\ldots + v_{PV} i_{PV}\, sin^2(\omega t)$$

Study of limit cycle oscillations (LCOs) in MPPT algorithms

Describing function of P&O MPPT

The time derivative of the instantaneous power is immediately found to be:

$$\boxed{\dot{P}_{PV}(t) = \omega\left(\bar{I}_{PV} v_{PV} + \bar{V}_{PV} i_{PV}\right)\cos(\omega t) + \omega i_{PV} v_{PV}\, sin(2\omega t)}$$

Considering that, in the vicinity of the MPP

$$\boxed{\frac{i_{PV}}{v_{PV}} = g_{eq}, \qquad \bar{I}_{PV} + g_{eq}\bar{V}_{PV} \cong 0,}$$

We can find:

$$\boxed{\dot{P}_{PV}(t) = \frac{\omega i_{PV}^2}{g_{eq}}\, sin(2\omega t)}$$

Study of limit cycle oscillations (LCOs) in MPPT algorithms

Describing function of P&O MPPT

We can now find an expression for the argument of the integral term in our continuous time model, i.e.

$$sign\left(\dot{P}_{PV}(t)\cdot \dot{I}_{PV}(t)\right) = sign\left(\frac{2\omega^2 i_{PV}^3}{g_{eq}} sin(\omega t)cos^2(\omega t)\right) = -sign(sin(\omega t))$$

The above function has a fundamental component that is opposite in phase with respect to the MPPT block input signal, i.e. the time varying component of I_{PV}.

Consequently, the integral function always generates an output signal whose fundamental component is *270° phase lagging the MPPT block input signal.*

Study of limit cycle oscillations (LCOs) in MPPT algorithms

Describing function of P&O MPPT

Computing the fundamental frequency Fourier coefficient of the time integral of the above expression, we can find the MPPT function output signal's fundamental harmonic amplitude. This turns out to be given by:

$$\boxed{D(j\omega) = \frac{4}{\pi}\frac{\Delta D}{\omega T_{MPPT}}}$$

The describing function of the non linear MPPT block, that is generally defined as the ratio between the first order Fourier coefficients of the output and input signals, can finally be found as:

$$\boxed{\Psi(i_{PV}) = \frac{D(j\omega)}{i_{PV}} e^{-j\frac{3}{2}\pi}}$$

Study of limit cycle oscillations (LCOs) in MPPT algorithms

Describing function of P&O MPPT

Finally, the explicit expression of the describing function for the P&O MPPT can be found to be given by:

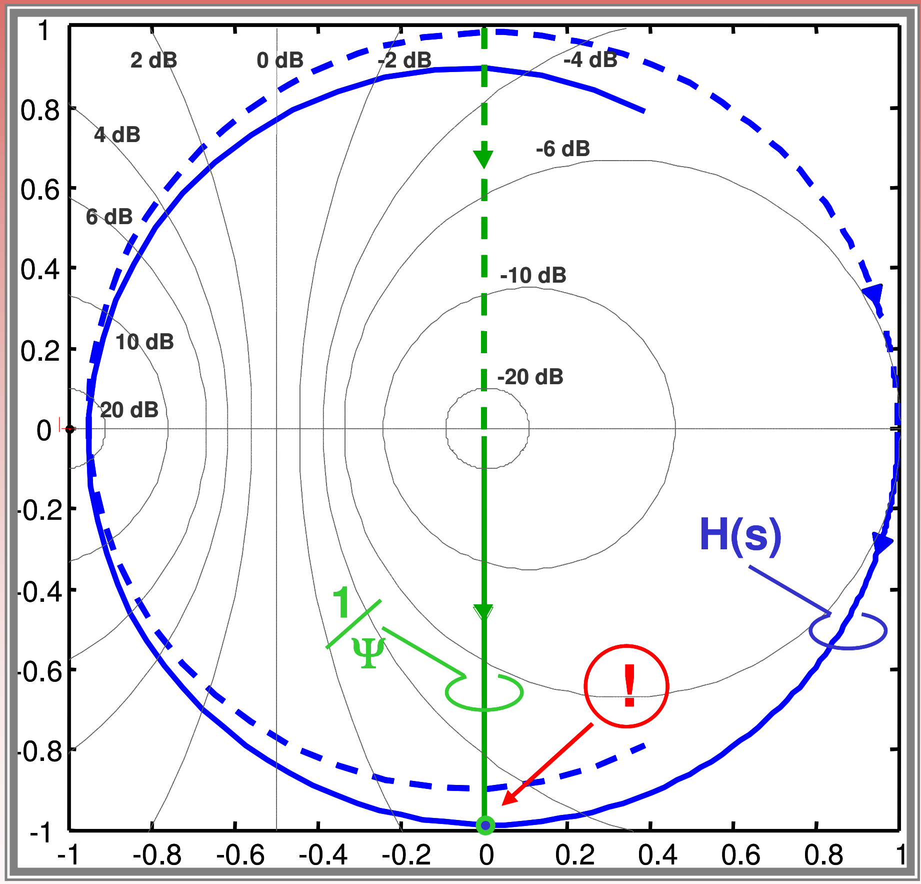
$$\Psi(i_{PV}) = \frac{4}{\pi} \frac{\Delta D}{\omega T_{MPPT}} \frac{e^{-j\frac{3}{2}\pi}}{i_{PV}}$$

It is now possible to use the describing function to evaluate the loop stability and to derive conditions for the existence of persistent oscillations, i.e. limit cycles, propagating around the loop of Fig. 6. In basic terms, a persistent oscillation will be free to propagate around the loop, if and only if it is possible to find a frequency, ω_{LCO}, where:

$$\Psi(i_{PV})H(j\omega_{LCO}) = 1$$

Study of limit cycle oscillations (LCOs) in MPPT algorithms

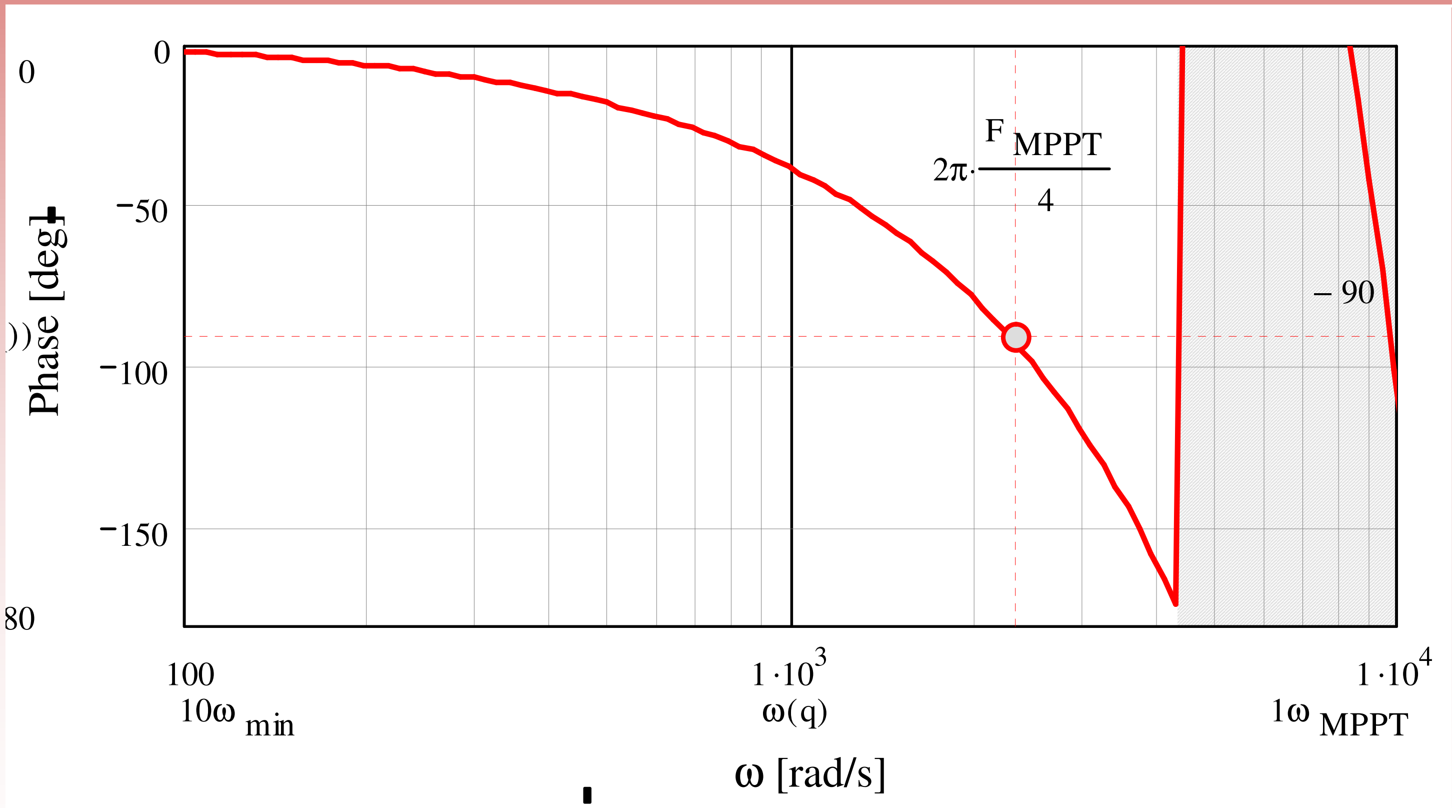
The above condition has a simple graphical interpretation in the Nyquist plane: in order to be satisfied, it requires the existence of an intersection point between the $H(j\omega)$ and the $1/\Psi(i_{PV})$ plots.



Please note that, for the sake of clarity, the $H(j\omega)$ plot has been stopped at a frequency that corresponds to a maximum phase rotation lower than 360°.

Study of limit cycle oscillations (LCOs) in MPPT algorithms

The frequency of the limit cycle will be the one where the linear plant *H(s)* introduces a phase rotation of -90°, that, summing to the non linear block equivalent phase lag of 270°, makes possible the propagation of a stable oscillation around the loop. Given the characteristics of *H(s)*, this frequency corresponds to $f_{LCO} = F_{MPPT}/4$.



Study of limit cycle oscillations (LCOs) in MPPT algorithms

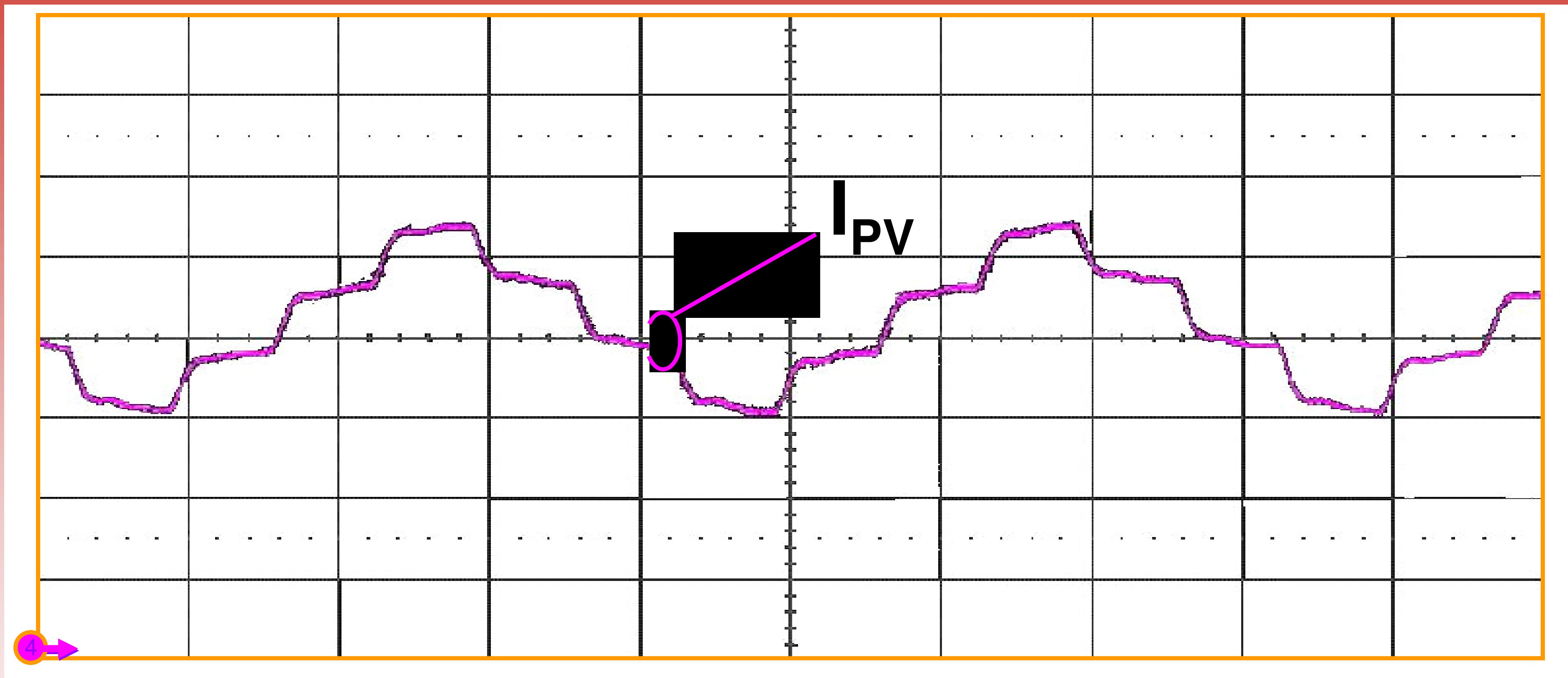
The situation above is typical of all cases where the MPPT algorithm iteration frequency is significantly, i.e. at least 50 times, lower than the modulation frequency. Similar plots can be obtained for lower MPPT frequencies, which means that, at low F_{MPPT} frequencies, the limit cycle oscillation will necessarily take place at a frequency $f_{LCO} = F_{MPPT}/4$. This result is in perfect agreement with practical experience.

Increasing the MPPT algorithm iteration frequency, the critical -90° phase rotation can be achieved at a progressively lower (i.e. lower than *1/4*) *relative frequency*. This is due to an increased contribution to the total loop phase lag of the current control loop and input capacitor.

As the physical system can oscillate only at even submultiples of the MPPT algorithm iteration frequency, the limit cycle oscillation will, at some point, take place at $f_{LCO} = F_{MPPT}/6$.

For F_{MPPT} values in between the LCO's behavior is generally unpredictable. The two oscillation frequencies are both visible, depending on the particular operating point and MPPT frequency.

Study of limit cycle oscillations (LCOs) in MPPT algorithms



Measured limit cycle oscillation of the P&O MPPT at F_{MPPT} = 1.5 kHz: the plot represents the PV generator's current, I_{PV} (100 mA/div, 1 ms/div). The DI_{REF} value amounts to 12 LSBs.

Study of limit cycle oscillations (LCOs) in MPPT algorithms

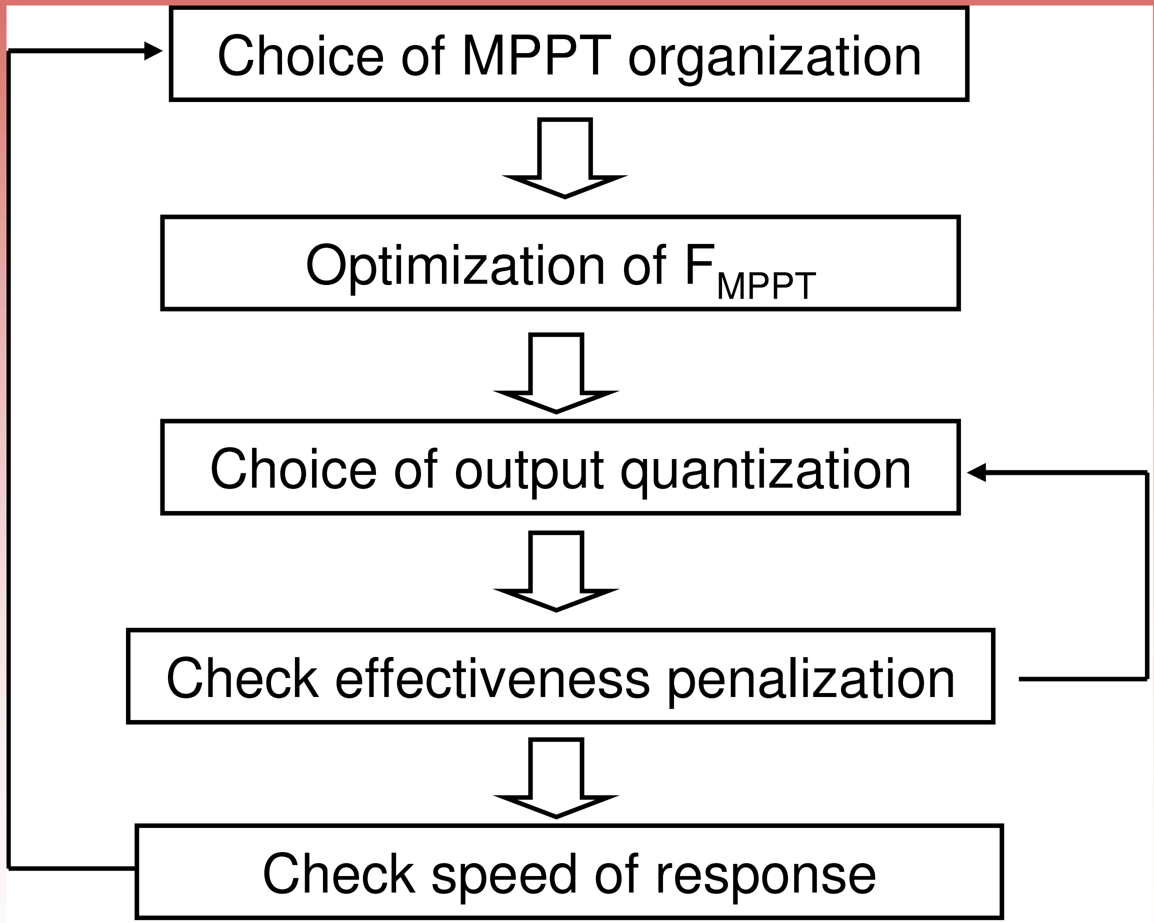
Based on the above considerations, it is possible to determine, for any organization of the digital controller, the maximum MPPT frequency that does not cause the amplification of the LCO, i.e. that maintains the critical phase rotation at the $F_{MPPT}/4$ frequency.

To complete the design of the MPPT algorithm, the output variable quantization has to be defined. Therefore, a new optimization has to be performed, trading-off the MPPT speed of response, that increases as Δd, ΔI_{REF} or ΔV_{REF} increases, and limit cycle amplitude, that increases as well. In order to keep the speed of response acceptable, it is normally required that the full output variable range is covered by a reduced number of increments, i.e. that N is not so high. For this reason, the output variable resolution is typically kept below 10 bits.

It is well known that, at the MPP, the PV generator voltage is much less sensitive to illumination conditions than its current. This means that, for a given range of irradiation conditions, e.g. from 100 W/m^2 to 1 kW/m^2, the voltage at the MPP is expected to vary relatively much less than the irradiation. Therefore, the MPPT based on current control is the most difficult to optimize. Either resolution is too coarse (at low irradiation) or speed of response is too low (at high irradiation levels).

Study of limit cycle oscillations (LCOs) in MPPT algorithms

Design procedure for the MPPT in a PV system



Digital current mode control: PI current regulator

We are now ready to discuss a simple digital current control algorithm implementation and the related design criteria.

The considered algorithm is the digital equivalent of an analog PI regulator.

The discussion will refer to an ideal controller implementation, where the above mentioned quantization effects can be considered negligible.

We assume the internal representation of variables is fractional, i.e. the full scale ranges of the AD converter and of the digital modulator are internally represented as unity.

It is important to underline that the normalization factors play a role in the definition of the loop gains. Different normalization factors imply different gains for the AD converter and for the DPWM modulator.

Digital current mode control: PI current regulator

Notes about the internal representation of variables

In the digital domain, signals are represented by sequences of binary numbers. In general, the transformation of an analog signal into a binary number requires the choice of a normalization factor. Let's discuss this issue with more detail.

The AD converter implicitly represents the converted data as unsigned fractions of its FSR.

As an example, an 8 bit ADC with a 5V FSR, would represent a 2.5V input signal as the exadecimal quantity 0x80, a 5V input signal as the exadecimal quantity 0xFF.

Almost always, these quantities will be processed by a fixed point calculator.

With this hardware, the decimal point is kept at a fixed position during calculations and cannot be dynamically adapted to the magnitude of data.

Digital current mode control: PI current regulator

Notes about the internal representation of variables

Therefore, the binary numbers generated by the ADC can be treated as integer quantities or fractional quantities, depending on the pre-defined position of the decimal point.

In general, in a n bit fixed point processor the internal representation of data will have the following format:

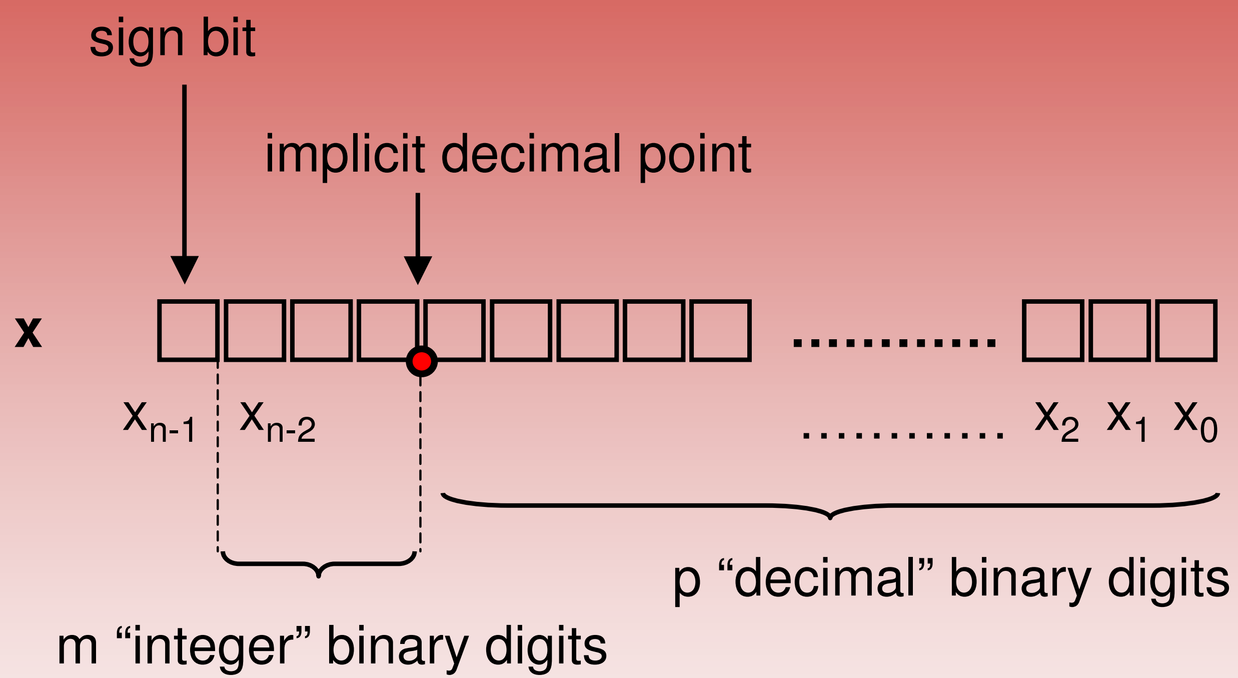
x = m.p, with m + p = n-1 (the MSB is used as a sign bit)

This means that a generic quantity *x* can be represented as a fractional number having *m* binary digits before and *p* binary digits after the decimal point.

When $p = 0$ the representation is said to be *integer,* when $m = 0$ it is said to be purely *fractional.* In the former case, the decimal point position is set to the right of the LSB, in the latter it is set to the right of the MSB (i.e. the sign bit).

Digital current mode control: PI current regulator

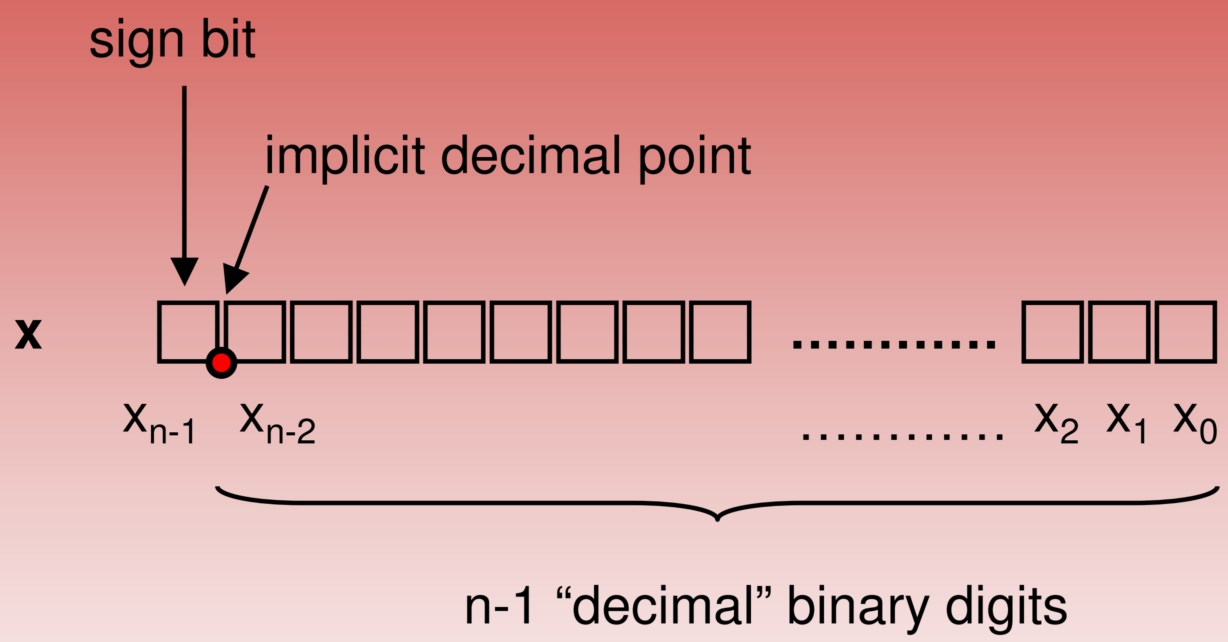
Notes about the internal representation of variables



Generic *m.p signed* fractional internal representation of a *n* bit binary number *x*.

Digital current mode control: PI current regulator

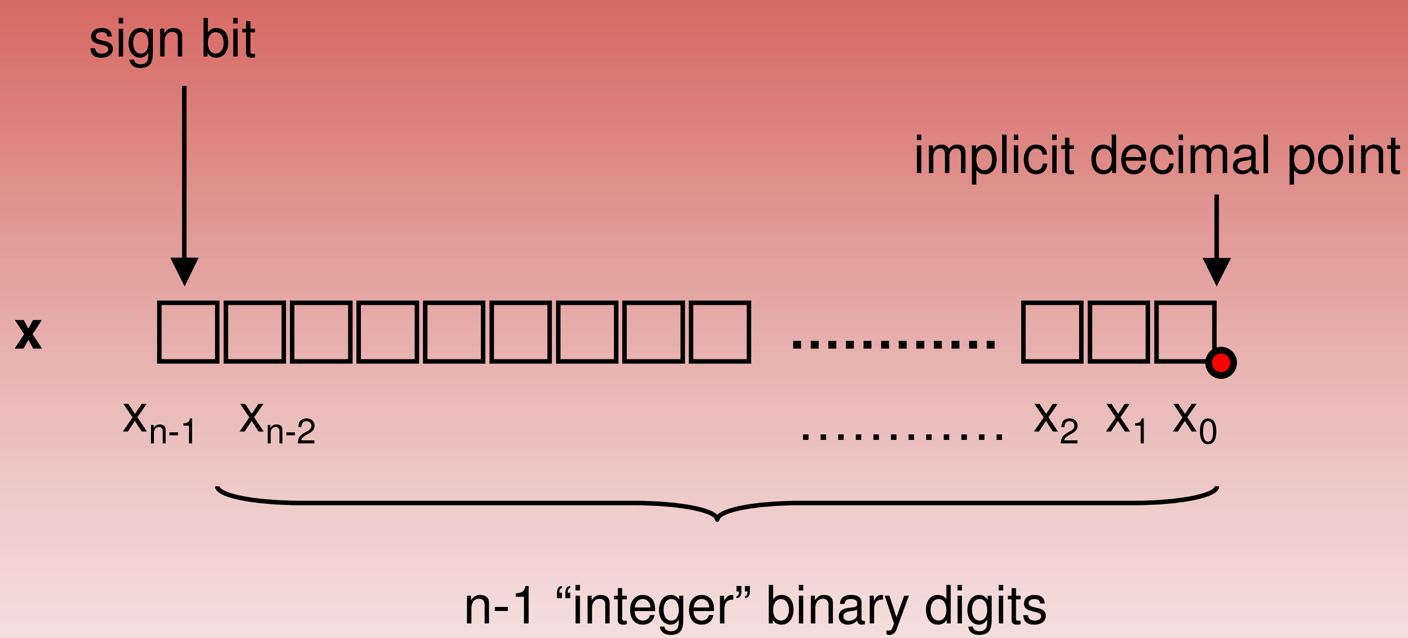
Notes about the internal representation of variables



Signed fractional (Qn-1) internal representation of a *n* bit binary number *x*.

Digital current mode control: PI current regulator

Notes about the internal representation of variables



Signed integer internal representation of a *n* bit binary number *x*.

Digital current mode control: PI current regulator

Notes about the internal representation of variables

The fractional normalization, also known as Qn-1 normalization (e.g. Q15, Q7), is often used because it makes the overflow after multiplication impossible.

Indeed, the multiplication of fractional quantities always gives a fractional result. Therefore, it is not possible to obtain, by multiplication of fractional quantities a result that falls outside the available range of numbers.

Please note that this is not true for generic m.p representations. In general, the result will require 2m bits for the integer part and, as such, might possibly fall outside the available range of numbers.

However, with fractional quantities, the position of the decimal point is implicitly shifted by any multiplication and has to be re-adjusted afterwards.

Digital current mode control: PI current regulator

Notes about the internal representation of variables

Indeed, multiplying two fractional quantities, x and y, implicitly shifts the decimal point to the right by one position. Therefore, to get back the correct scaling, the result has to be left shifted by one bit. Consider the example:

$$\underbrace{x \cdot 2^{n-1}} \cdot \underbrace{y \cdot 2^{n-1}} = x \cdot y \cdot 2^{2n-2}$$

Integer equivalents of x and y

The result (that occupies 2n bits) has the decimal point two bits to the right of the MSB!

The correct scaling factor is *2n-1*, so we have to shift the result *left by 1 bit* to align it with the other variables. Some microcontrollers and DSPs allow to implement the fractional multiplication and to adjust the decimal point position automatically, without any intervention of the programmer.

Digital current mode control: PI current regulator

Notes about the normalization of variables

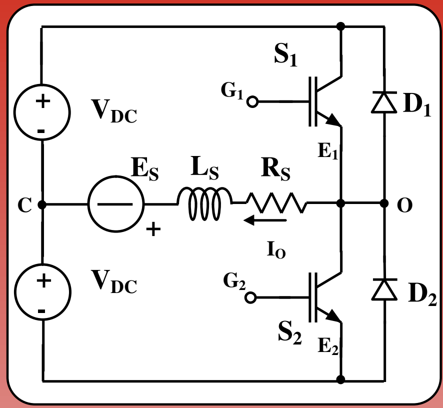
When fractional normalization is used, the ADC converter gain is implicitly set to 1/FSR, as an input signal equal to the FSR generates the maximum internally representable fraction, that is unity (or, actually, $1 - 2^{-n+1}$).

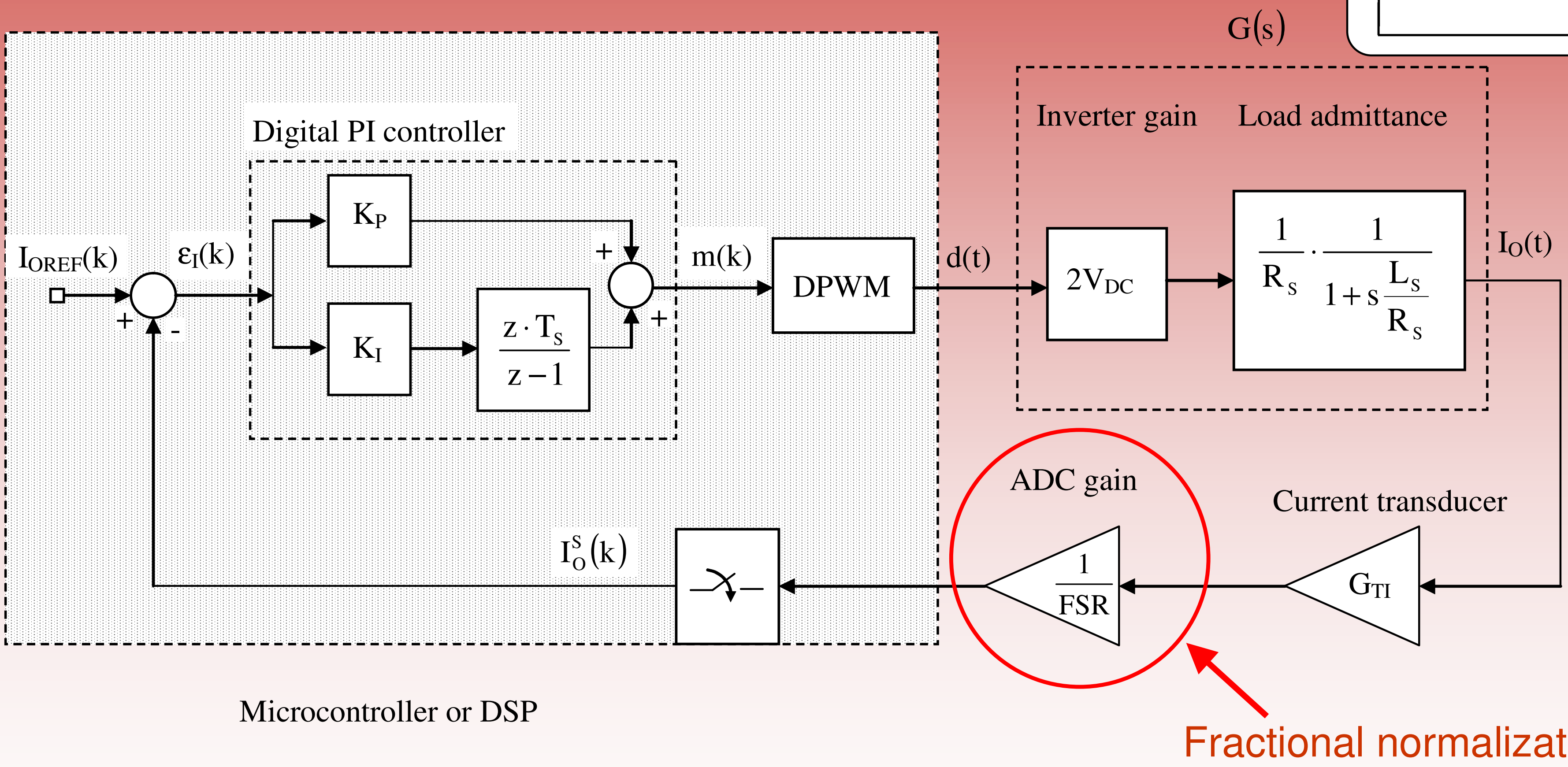
In addition, the PWM modulator gain is automatically set to unity, as the maximum internally representable fraction is turned into a unity duty-cycle.

Of course, if different normalization factors were used, the equivalent gains for the ADC and the DPWM would have to be adjusted.

Digital current mode control: PI current regulator

Model of the sampled data system





Digital PI current controller organization: a sampled data system.

Digital current mode control: PI current regulator

Model of the sampled data system

The controller and modulator blocks are inside the digital domain, the shaded area, that represents a microcontroller or DSP board.

The inverter and transducer models are instead specified as continuous time models.

The link between the two time domains is represented by the ideal sampler at the input of the controller and by the digital pulse width modulator, that generates the controller output and inherently implements the interpolator, or holder, function.

All these characteristics imply that we are dealing with a *sampled data* dynamic system.

We assume the controller operation is clocked by the DPWM, i.e. a new iteration of the control algorithm is started as soon as a modulation period begins. We also assume that the single update mode of operation is adopted.

Digital current mode control: PI current regulator

Model of the sampled data system

It can be convenient to derive the digital controller from an existing analog design or to exploit previous experience in continuous time control, applying the same techniques to the sampled data system's case.

This motivates a procedure, that is called *controller discretization,* that has the advantage of requiring only a minimal knowledge of digital control theory to be successfully applied.

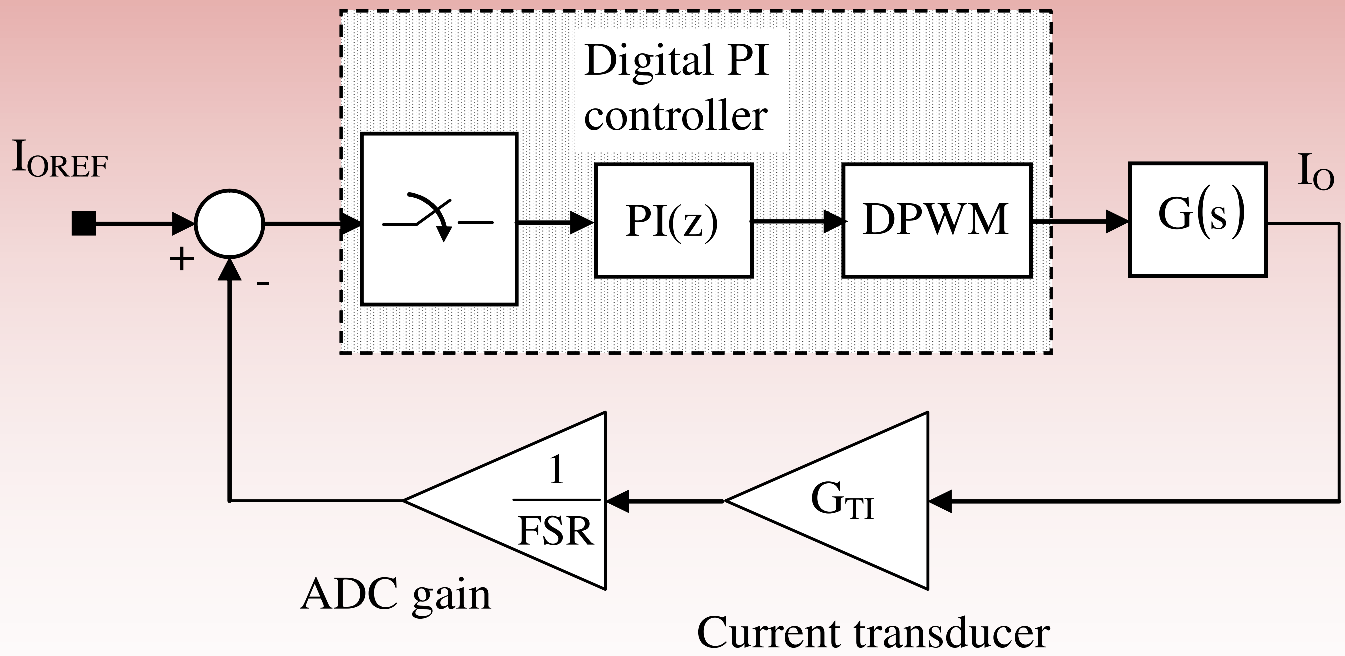
All that is needed is a satisfactory analog controller design and the application of one of the several possible discretization methods to turn the analog controller into a digital one.

As we will see in the following, although generally satisfactory, the application of this method implies some loss of precision, as compared to a direct digital design, mainly due to the approximations involved in the discretization process itself and in the equivalent continuous time representation of delays.

Digital current mode control: PI current regulator

Model of the sampled data system

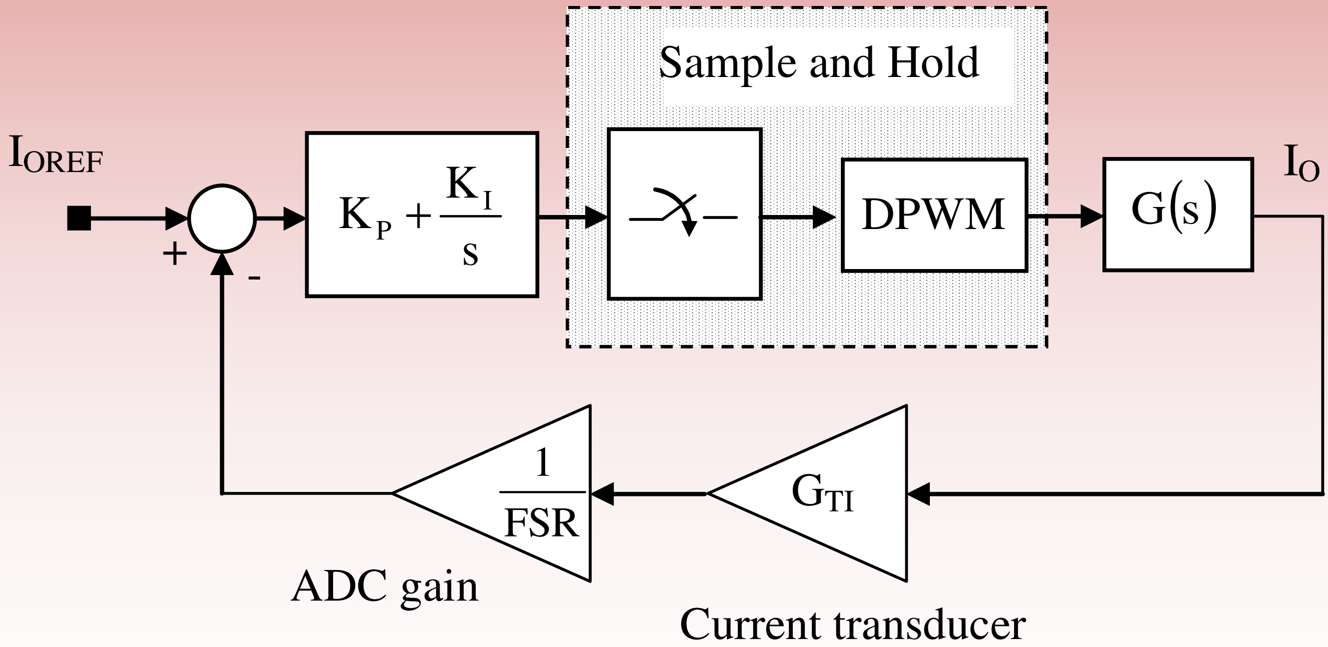
The continuous time synthesis of a controller for a sampled data dynamic system requires to model, in the continuous time domain, the discrete time system included between the ideal sampler located at the controller input and the output interpolator.



Digital current mode control: PI current regulator

Model of the sampled data system

The typical way to do this is considering a suitable model of the interpolator, e.g. some kind of holder, and, after that, finding an equivalent continuous time representation for the cascade connection of the ideal sampler and the holder, that is called a *sample and hold.* Please note that this method is actually what we have already used in modelling the different types of DPWM.



Digital current mode control: PI current regulator

Model of the sampled data system

Once the sample and hold is modelled, the designer can operate the controller synthesis in the continuous time domain, assuming that, once converted *back into a discrete time equivalent* and inserted between the sampler and the interpolator in the original sampled data system, the controller will maintain the closed loop properties determined by the continuous time design.

In our case, the function of the interpolator is inherent to the DPWM, because that is the block where the conversion from the digital to the analog domain takes place.

This means that, once the holder effect is properly modelled in the DPWM, the conversion of the sampled data system into an equivalent, continuous time one will be complete. This may seem a minor detail, but in this difference lies the key for the correct interpretation of our design procedure.

Digital current mode control: PI current regulator

Model of the sampled data system

Considering, for example, the model for the symmetric pulse modulator, after minor re-arrangements and assuming, as we explained before, unity gain for the modulator i.e. $c_{PK} = 1$, we get the following expression:

$$\mathrm{DPWM}(s) = \frac{1}{2}\left(e^{-s(1-D)\frac{T_S}{2}} + e^{-s(1+D)\frac{T_S}{2}} \right) = e^{-s\frac{T_S}{2}} \cos\left(\omega \frac{T_S}{2} D \right) \cong e^{-s\frac{T_S}{2}}$$

where the last approximation holds as long as we can consider the typical current controller bandwidth to be limited well below the modulation frequency, $1/T_S$. Indeed, under this assumption, the gain term can actually be approximated by unity, independently from the duty-cycle D.

Therefore, the model predicts for the DPWM a pure delay behavior, with half a modulation period duration. This is exactly equal to a ZOH model!

Digital current mode control: PI current regulator

Model of the sampled data system

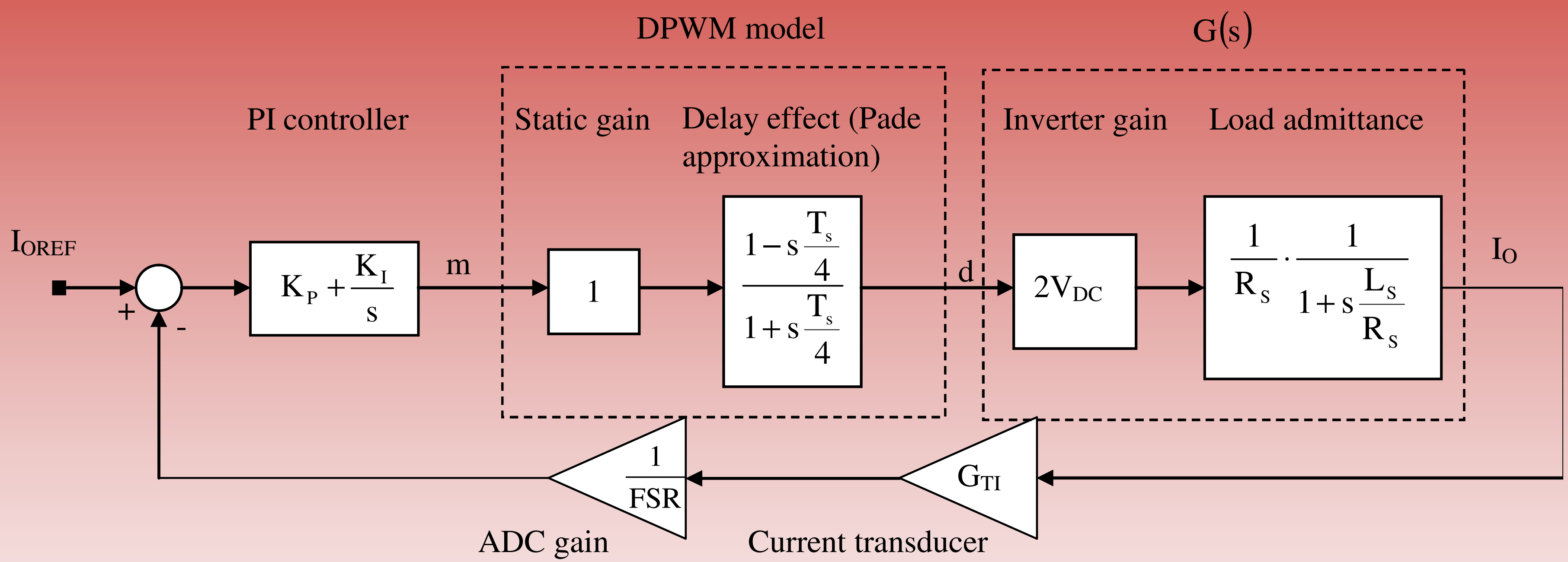
In order to simplify the controller design equations, a first order approximation of the delay effect is sometimes taken into account, according to the following relation:

$$e^{-s\frac{T_S}{2}} \cong \frac{1 - s\frac{T_S}{4}}{1 + s\frac{T_S}{4}}$$

With this we have completed the continuous time model of the sample and hold block. We can therefore design the continuous time PI controller for the current control loop.

Digital current mode control: PI current regulator

Model of the sampled data system



It is interesting to observe that the model we obtained differs from the one we would have adopted for a continuous time system only for the delay effect induced by the digital modulator.
A consequence of this is that the design we are going to obtain, compared to a purely analog one, will necessarily be *more conservative.*

Digital current mode control: PI current regulator

The design of the PI controller in the continuous time domain is straightforward. We only need to set the desired crossover frequency, ω_{CL}, that will approximate the closed loop regulator's bandwidth and a suitable phase margin, ph_m, to provide a sufficient dampening action. These requirements determine the following design equations:

$$\begin{cases} \dfrac{K_I}{K_P} = \dfrac{\omega_{CL}}{\tan\left(-90° + ph_m + 2\tan^{-1}\left(\omega_{CL}\dfrac{T_S}{4}\right) + \tan^{-1}\left(\omega_{CL}\dfrac{L_S}{R_S}\right)\right)} \\ K_P = \dfrac{FSR}{2V_{DC}} \cdot \dfrac{R_S}{G_{TI}} \cdot \sqrt{\dfrac{1+\left(\omega_{CL}\dfrac{L_S}{R_S}\right)^2}{1+\left(\dfrac{1}{\omega_{CL}}\dfrac{K_I}{K_P}\right)^2}} \end{cases}$$

Digital current mode control: PI current regulator

Half Bridge Inverter Parameters

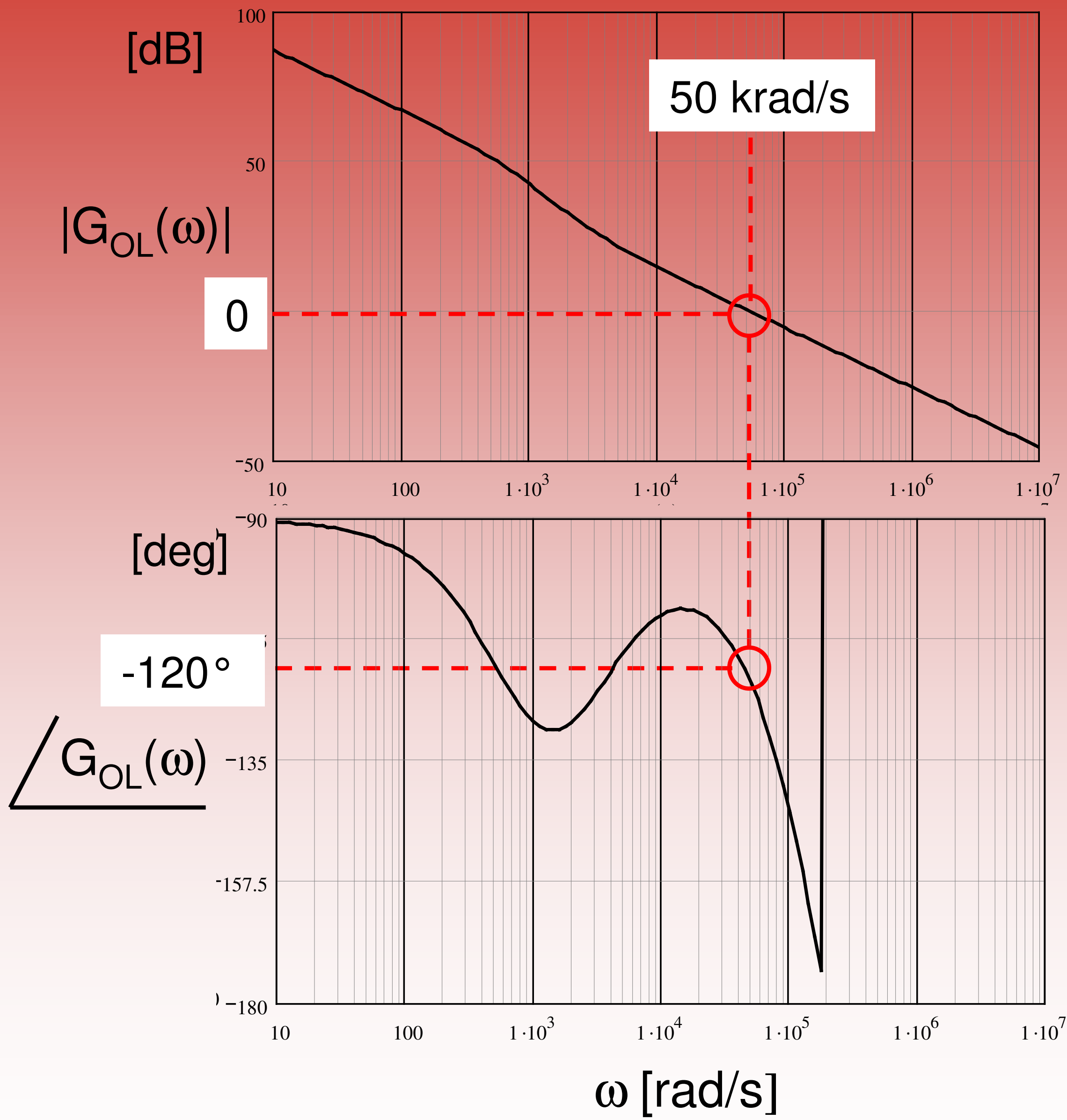
Rated ouput power, P_O	1000	[W]
Phase inductance, L_S	1.5	[mH]
Phase resistance, R_S	1	[W]
Phase load voltage, E_S	100	[V_{RMS}]
Load frequency, f_O	125	[Hz]
DC link voltage, V_{DC}	250	[V]
Switching frequency, f_S	50	[kHz]
ADC full scale range, FSR	4	[V]
Current transducer gain, G_{TI}	0.1	[V/A]

Digital current mode control: PI current regulator

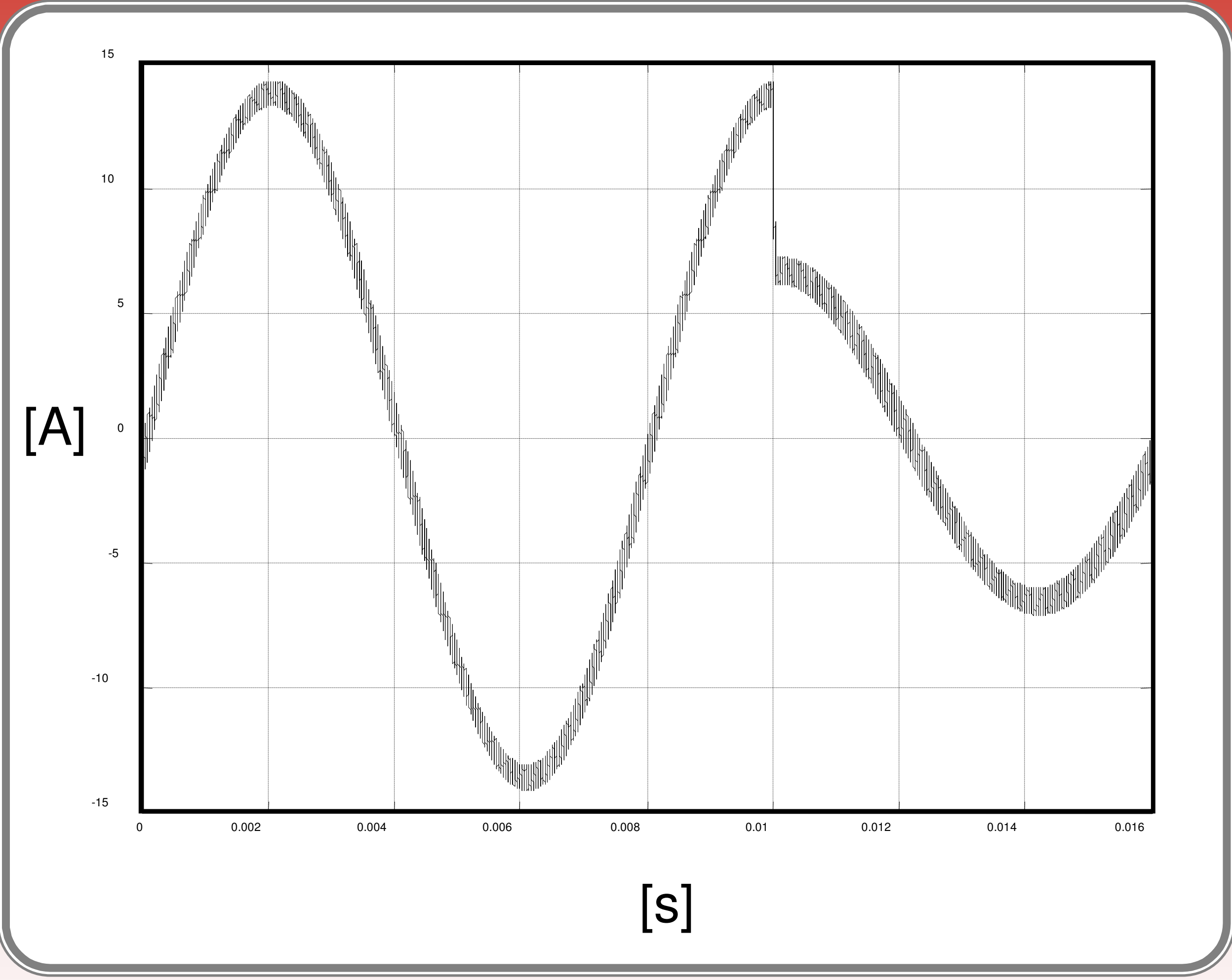
With the above parameters and considering a crossover frequency equal to *1/6* of the switching frequency and a *60°* phase margin, we get:

$$K_P = 6.274, \quad K_I = 1.8 \cdot 10^4 \text{ [rad/s]}$$

from which, in turn, we obtain this Bode diagram of the open loop gain $G_{OL}(\omega)$.

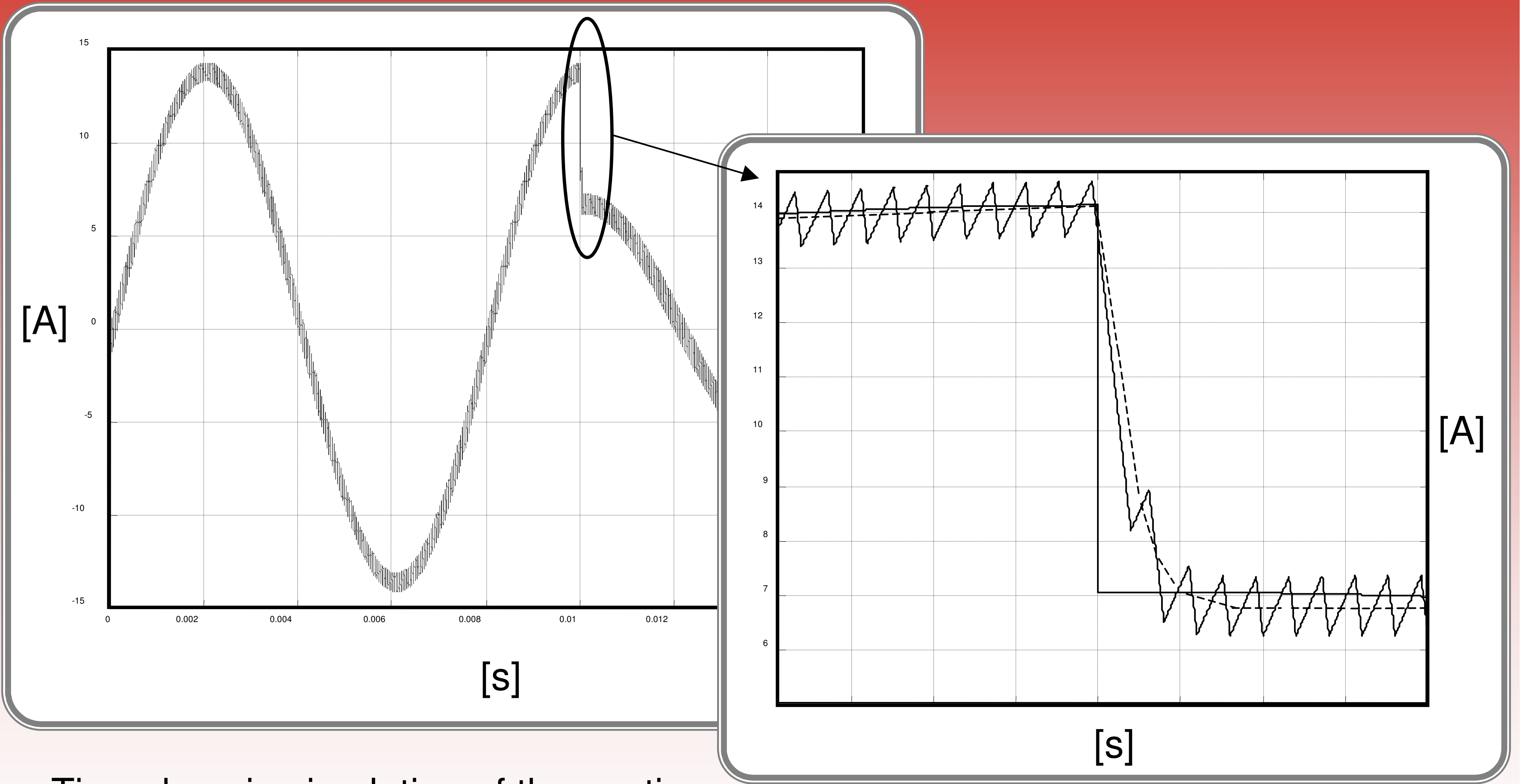


Digital current mode control: PI current regulator



Time domain simulation of the continuous time system. Response to a step reference change.

Digital current mode control: PI current regulator



Time domain simulation of the continuous time system. Response to a step reference change.