Digital control of switching mode power supplies

Simone Buso

University of Padova – ITALY

Dept. of Information Engineering – DEI

Lesson 3

Digital proportional integral controller

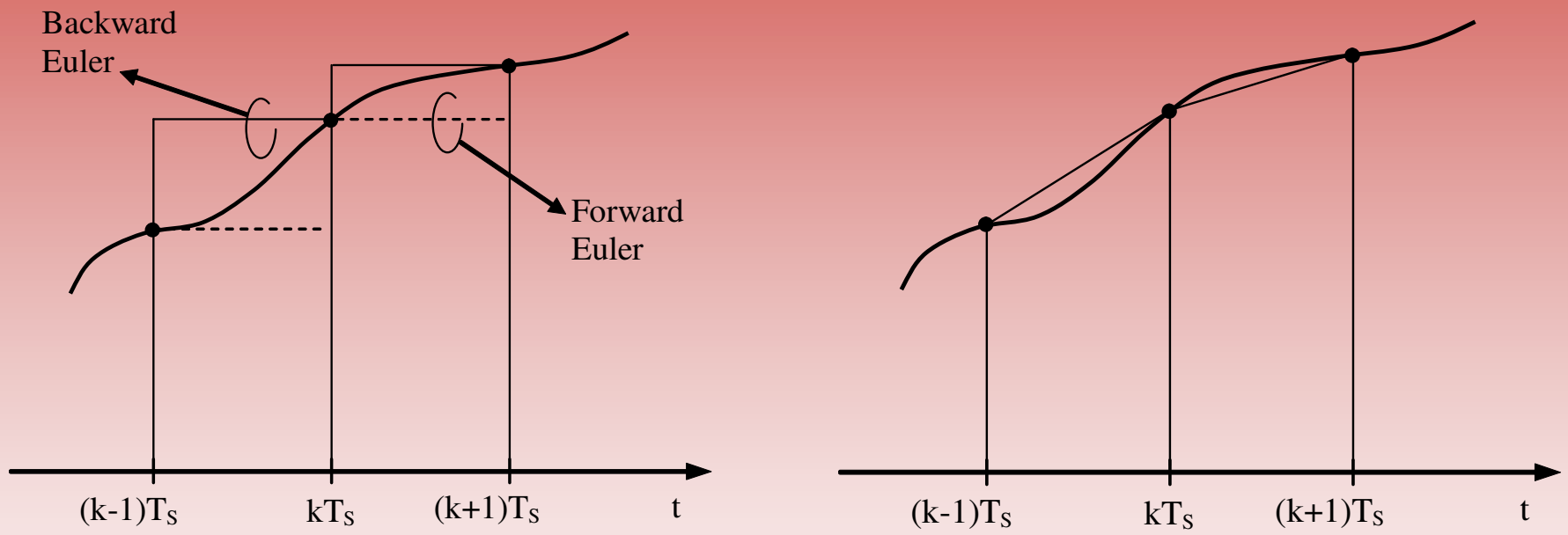
Continuous time controller discretization strategies
Effects of the computation delay
Derivation of a discrete time domain converter dynamic model

Digital predictive (dead beat) controller

Physical approach to the derivation of the dead-beat controller
State space approach
Basic robustness analysis of the dead beat controller

Digital current mode control: PI current regulator

Discretization strategies: Euler and trapezoidal integration



Examples of numerical integration methods. Euler (left) and trapezoidal (right)

Digital current mode control: PI current regulator

Discretization strategies: Euler and trapezoidal integration

The basic concept behind these discretization methods is very simple: we want to replace the continuous time computation of integrals with some form of numerical approximation. The two basic methods that can be applied to this purpose are known as Euler integration and trapezoidal integration method.

As can be seen, the area under the curve is approximated as the sum of rectangular or trapezoidal areas. The Euler integration method can actually be implemented in two ways, known as *forward* and *backward* Euler integration, the meaning being obvious.

Writing the rule to calculate the area as a recursive function of the signal samples, applying Z-transform to this area function, and imposing the equivalence with the Laplace transform integral operator, gives a direct transformation from the Laplace transform independent variable *s* to the Z-transform independent variable *z*.

Digital current mode control: PI current regulator

Discretization strategies: Euler and trapezoidal integration

Method	Z-form	3% distortion limit
Backward Euler	$s = \frac{z-1}{z \cdot T_s}$	$\frac{f_s}{f} > 20$
Forward Euler	$s = \frac{z-1}{T_s}$	$\frac{f_s}{f} > 20$
Trapezoidal (Tustin)	$s = \frac{2}{T_s}\frac{z-1}{z+1}$	$\frac{f_s}{f} > 10$

Digital current mode control: PI current regulator

Discretization strategies: Euler and trapezoidal integration

Since the numerical integration methods imply a certain degree of approximation, if we compare the frequency response of the controller before and after discretization, some degree of distortion, also known as frequency warping effect, can always be observed.

The table also shows the condition that has to be satisfied to make the distortion lower that 3% at a given frequency *f*. The condition is expressed as a limit for the ratio between the sampling frequency $f_S = 1/T_S$ and the frequency of interest, *f*.

As can be seen, the trapezoidal integration method, that generates the so-called Tustin Z-form, is more precise than the Euler method, and, as such, guarantees a smaller distortion at each frequency or, equivalently, an higher 3% distortion limit, that is as high as one tenth of the sampling frequency.

Digital current mode control: PI current regulator

Discretization strategies: Euler and trapezoidal integration

Ideally, it is also possible to pre-warp the controller transfer function so as to compensate the frequency distortion induced by the discretization method and get an *exact* phase and amplitude match of the continuous time and discrete time controllers at *one* given frequency, that is normally the desired crossover frequency. Nevertheless, the overall performance improvement is normally negligible.

Let's apply backward Euler's discretization to our PI current controller.

$$PI(s) = K_I \frac{1 + s \cdot \frac{K_P}{K_I}}{s}$$

Substituting the appropriate Z-form we get:

Digital current mode control: PI current regulator

Discretization strategies: Euler and trapezoidal integration

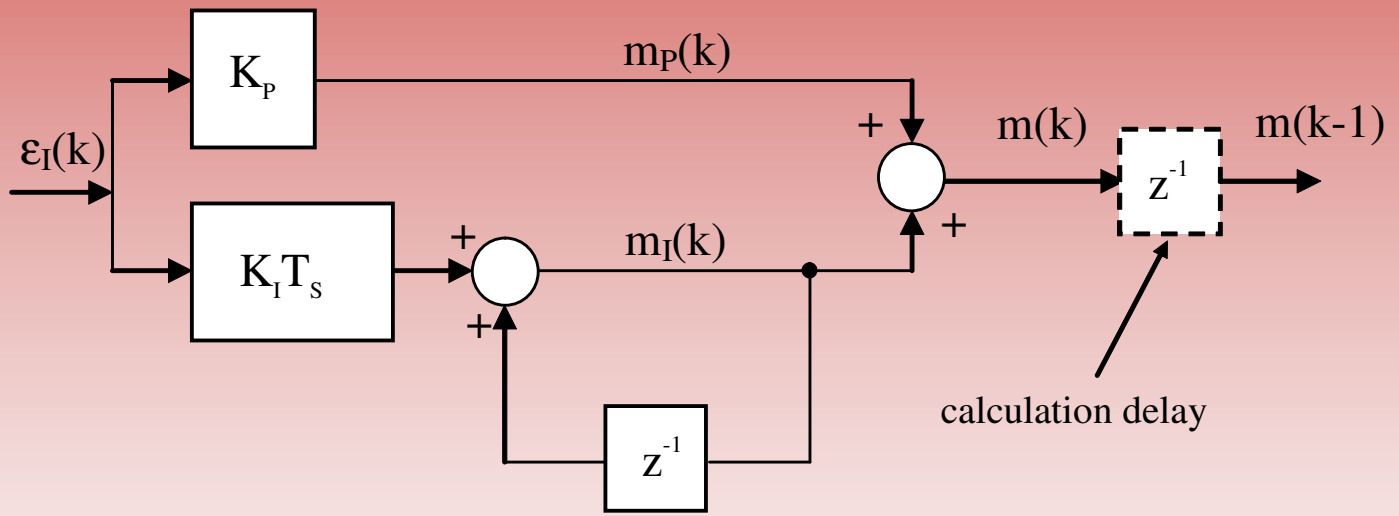
$$PI(z) = K_I \underbrace{\frac{1 + \frac{z-1}{z \cdot T_S} \cdot \frac{K_P}{K_I}}{\frac{z-1}{z \cdot T_S}}} = \frac{(K_P + K_I \cdot T_S) \cdot z - K_P}{z-1} = K_P + K_I \cdot T_S \cdot \frac{z}{z-1}$$

Discrete time integrator

The above rational transfer function can be simplified to give the discrete time implementation of the PI controller. As we can see, the final expression comprises a discrete time integrator, preserving the basic PI structure. A parallel implementation of the discrete time PI is immediate to derive.

Digital current mode control: PI current regulator

Discretization strategies: Euler and trapezoidal integration



Block diagram representation of the digital PI controller

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

The following issues need to be clarified with respect to the discrete time PI controller implementation:

- the use of different discretization strategies;
- the large signal behavior of the digital integrator;
- the role of the computation delay;

We will discuss all of them in the following.

The use of Tustin Z-form, determines a different rational transfer function fo the digital PI. The differences with respect to the previous result can be better put into evidence by transforming the PI mathematical expression into an algorithm, that can be directly programmed into any microcontroller or DSP.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

$$\begin{cases} m_I(k) = K_I \cdot T_S \cdot \varepsilon_I(k) + m_I(k-1) \\ m(k) = m_P(k) + m_I(k) = K_P \cdot \varepsilon_I(k) + m_I(k) \end{cases}$$

Euler based digital PI algorithm: please note that index *k* represents in a compact form the time instant kT_S, where T_S is the sampling period.

$$\begin{cases} m_I(k) = K_I \cdot T_S \cdot \dfrac{\varepsilon_I(k) + \varepsilon_I(k-1)}{2} + m_I(k-1) \\ m(k) = m_P(k) + m_I(k) = K_P \cdot \varepsilon_I(k) + m_I(k) \end{cases}$$

Tustin based digital PI algorithm.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

As can be seen, the structure of the two expressions is similar, the only difference being determined by the computation of the integral part that, in the Tustin PI algorithm, is not based on a single current error value, but rather on the moving average of the two most recent current error samples.

This relatively minor difference is responsible for the lower frequency response distortion of the Tustin transform.

It is worth noting that the proportional and integral gains for the two different versions of the discretized PI controller are exactly the same. As can be seen, in both cases we find that the proportional gain for the digital controller is exactly equal to that of the analog controller, while the digital integral gain can be obtained simply by multiplying the continuous time integral gain and the sampling period.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

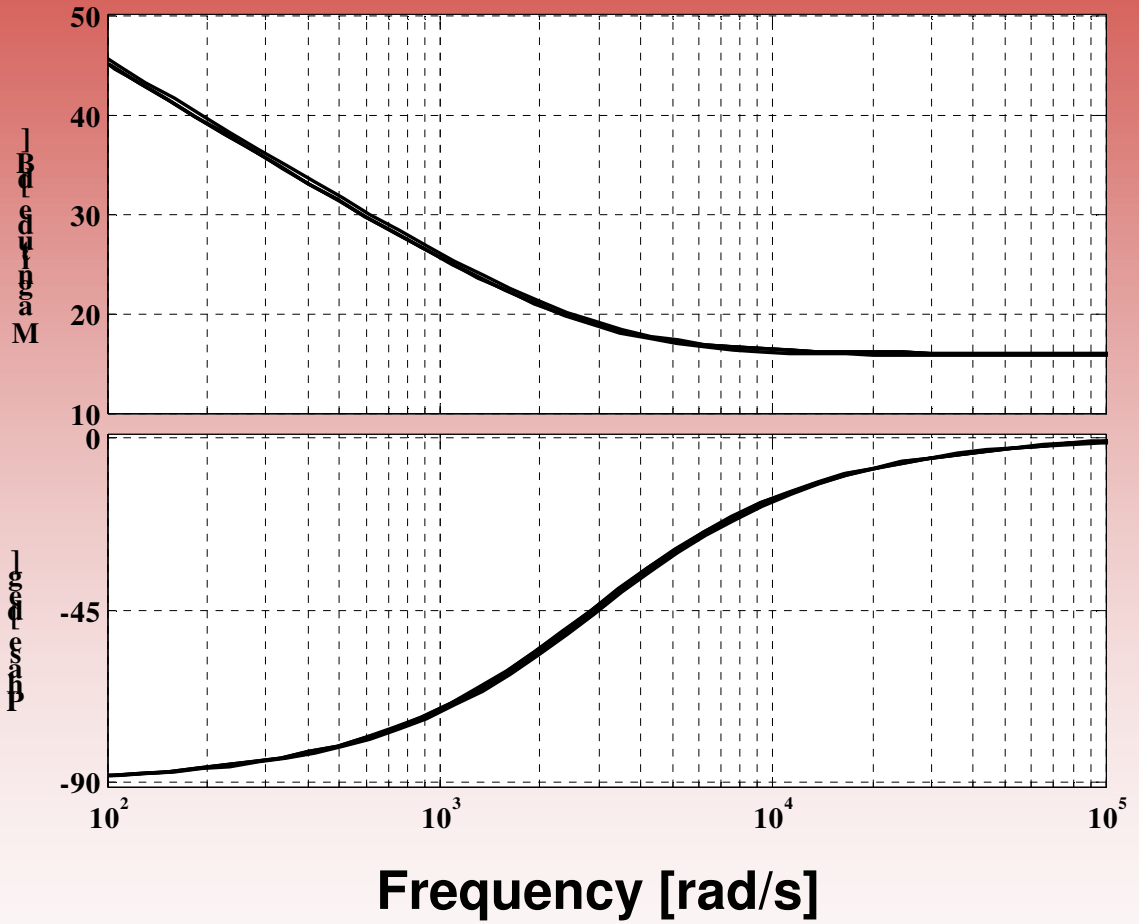
In summary, we have seen that, given a suitably designed analog PI regulator, the application of any of the considered discretization strategies simply requires the computation of the digital PI gains, as in the following:

$$\begin{cases} K_{I\_dig} = K_I \cdot T_S \\ K_{P\_dig} = K_P \end{cases}$$

and the implementation of the proper control algorithm. Note that even the application of pre-warping does not change much the values of the controller gains, especially when a relatively high ratio between the sampling frequency and the desired crossover frequency is possible.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues



Bode plots of the different PI realizations.

This is further confirmed by these Bode plots, that refer to. the original continuous time PI controller and to each one of its three discretized versions (Euler, Tustin and pre-warped). As can be seen, with our design parameters and sampling frequency, the plots are practically undistinguishable.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

The integral part wind-up phenomenon can take place any time the PI controller's input signal, i.e. the regulation error, is different from zero for relatively large amounts of time.

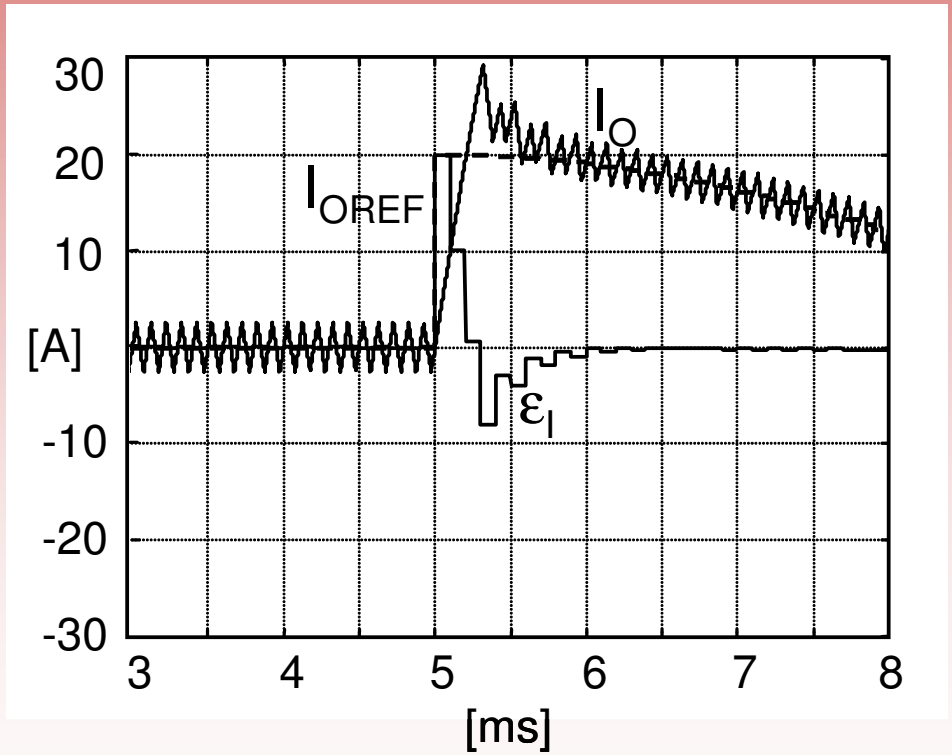
This typically happens in the presence of large reference amplitude variations or other transients, causing inverter saturation. The problem is determined by the fact that, if we do not take any countermeasure, the integral part of the controller will be accumulating the integral of the error for the entire transient duration.

Therefore, when the new set-point is reached, the integral controller will be very far from the steady state and a transient will be generated on the controller variable, that typically has the form of an overshoot.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

It is fundamental to underline that this overshoot is not related to the small signal stability of the system. Even if the phase margin is high enough, the transient will always be generated, as it is just due to the way the integral controller reacts to converter saturation.



Dynamic behaviour of the PI controller during saturation.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

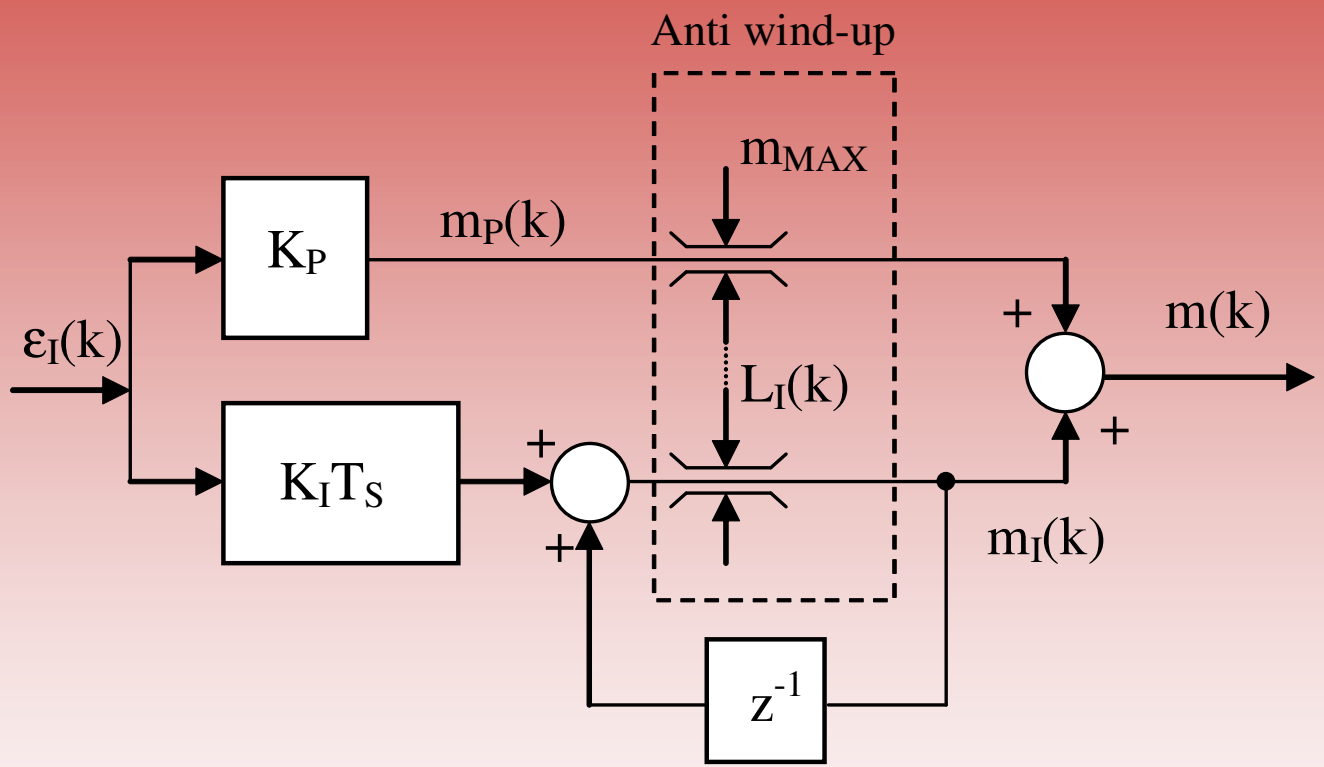
The solution to this problem is based on the dynamic limitation of the integral controller output during transients. Transients can be detected monitoring the output of the controller proportional part: in a basic implementation, any time this is higher than a given limit, the output of the integral part of the controller can be set to zero. Integration is resumed only when the regulated variable is again close to its set-point, i.e. when the output of the proportional part gets below the specified limit.

More sophisticated implementations of this concept are also possible, where the limitation of the integral part is done gradually, for example keeping the sum of the proportional and integral outputs in any case lower or equal than a predefined limit.

This implementation, of course, requires a slightly higher computational effort, that amounts to the determination of the following quantity, where m_{MAX} is the controller output limit: $|L_I(k)| = m_{MAX} - |K_p \varepsilon_I(k)|$.

Digital current mode control: PI current regulator

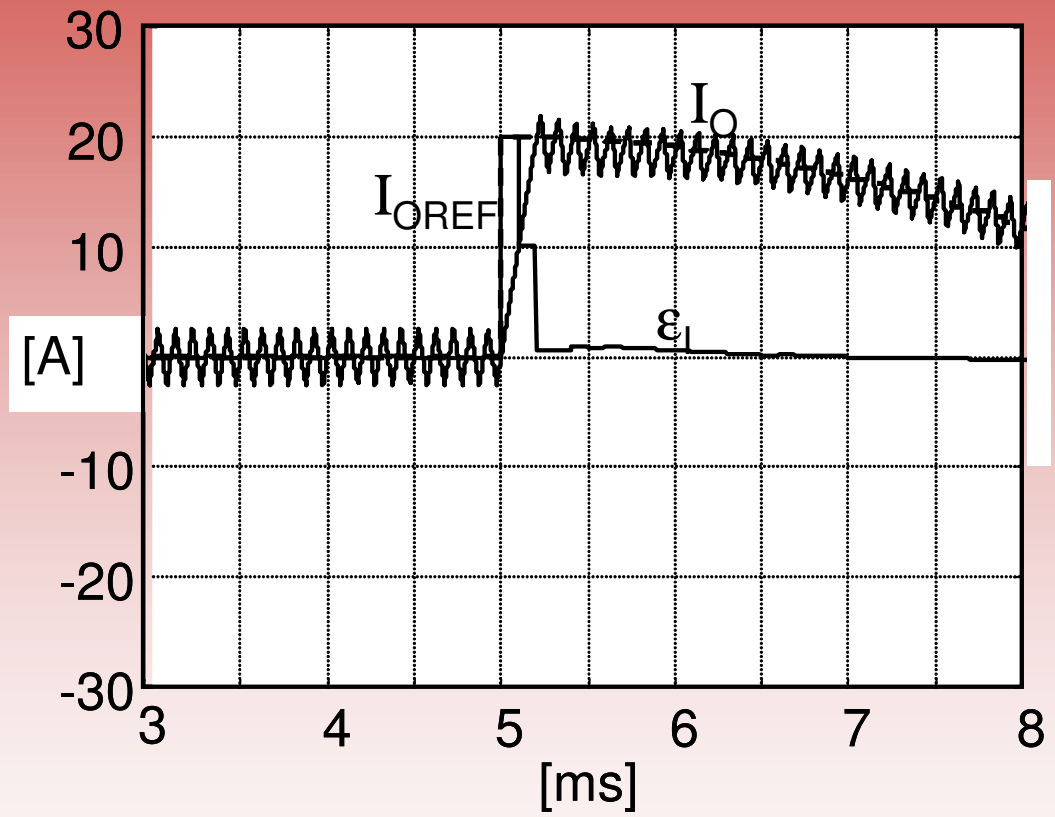
Discrete time PI current controller: implementation issues



Block diagram representation of the digital PI controller with anti wind-up action

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues



Dynamic behaviour of the PI controller during saturation with anti wind-up

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

In our discussion, we have shown how the delay effect associated to the DPWM operation can be taken care of. An additional complication we have to deal with is represented by the fact that the digital control loop actually hides a second, independent source of delay: this is the control algorithm computation delay, i.e. the time required by the processor to compute a new *m* value, given the input variable sample.

Although digital signal processors and microcontrollers are getting faster and faster, in practice, the computation time of a digital current controller always represents a significant fraction of the modulation period, ranging typically from 10% to 40% of it. A direct consequence of this hardware limitation is that, in general, we cannot compute the input to the modulator during the same modulation period when it has to be applied. In other words, the modulator input, in any given modulation period, must have been computed during the previous control algorithm iteration. Dynamically, this means that the control algorithm actually determines an additional one modulation period delay.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

One could consider this analysis to be somewhat pessimistic, because powerful microcontrollers and DSPs are available today, that allow the computation of a PID routine in much less than a microsecond.

However, it is important to keep in mind that, in industrial applications, the cost factor is fundamental: cost optimization normally requires the use of the minimum hardware that can fulfil a given task. The availability of hardware resources in excess, with respect to what is strictly needed, simply identifies a poor system design, where little attention has been paid to the cost factor.

Therefore, the digital control designer will struggle to fit his or her control routine to a minimum complexity microcontroller much more often than he or she will experience the opposite situation, where a high speed DSP will be available just for the implementation of a digital PI or PID controller.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

The conventional approach to tackle the problem consists in assuming a whole control period is dedicated to computations. In this case, in order to get from the digital controller a satisfactory performance, the calculation delay effect has to be included from the beginning in the analog controller design.

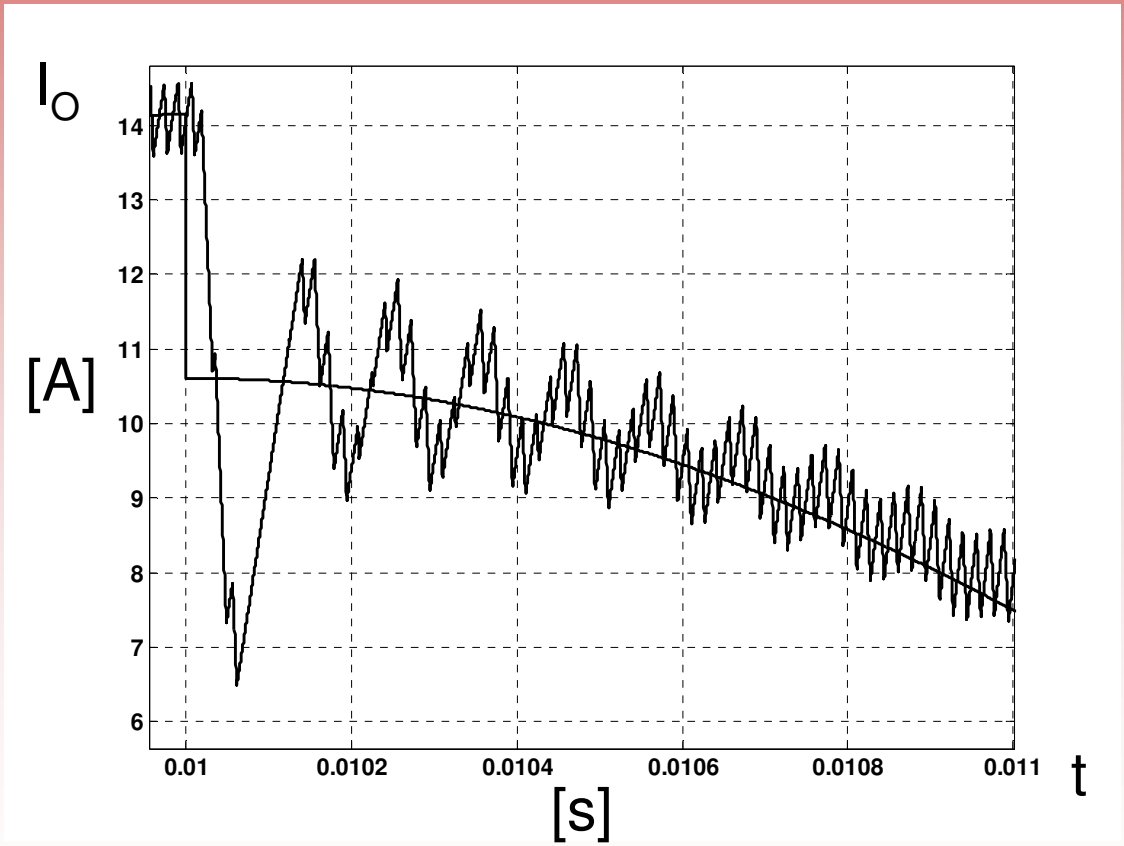
Practically, this can be done increasing the delay effect represented by the Padé approximation T_S. After that, the procedure for the controller synthesis through discretization can be re-applied.

It is important to underline once more that, if the analog controller is not re-designed and a significant calculation delay is associated to the implemented algorithm, the achieved performance can be much less than satisfactory.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

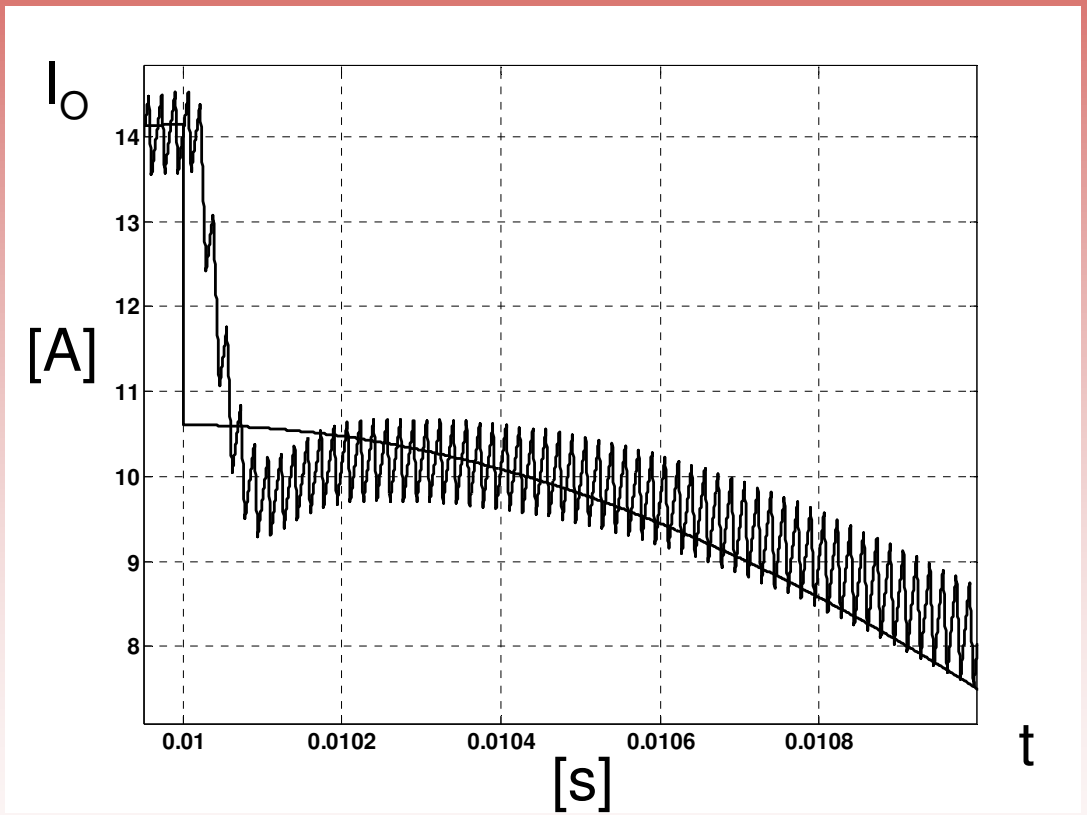
An example of this situation is shown in the figure below, where a calculation delay equal to one modulation period is considered. Note how the step response tends to be under-damped.



Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

In this case instead, the dynamic response of the re-designed controller is smoother, but a significant reduction of its speed can be observed.



Please note that the result has been obtained by reducing the crossover frequency to $f_S/15$, while keeping the same phase margin of the original design.

Digital current mode control: PI current regulator

Discrete time PI current controller: implementation issues

The previous example shows that, when the maximum performance is required, this conventional approach may be excessively conservative. Penalizing the controller bandwidth to cope with the computation delay, the synthesis procedure will unavoidably lead to a worse performance, with respect to conventional analog controllers.

This is the reason why, in some cases, a different modelling of the digital controller can be considered, that takes into account the exact duration of the computation delay and so, by using modified Z-transform, exactly models the duty-cycle update instant within the modulation period.

Doing this, a significant performance improvement can be achieved and the penalization of the digital controller with respect to the analog one can be minimized.

Digital current mode control: PI current regulator

Digital PI current controller: discrete time synthesis

What we have described so far is a very simple digital controller design approach. It is based on the transformation of the sampled data system into a continuous time equivalent, that is used to design the regulator with the well known continuous time design techniques.

The symmetrical approach is possible as well. In this case, the sampled data system is transformed into a discrete time equivalent, that can be used to design the controller directly in the discrete time domain.

A detailed and precise discrete-time converter model is generally based on the integration of the linear and time-invariant state space equations, associated to each switch configuration (i.e. turn-on and turn-off).

The state variable time evolutions, obtained separately for each topological or switch state, are linked to one another exploiting their continuity, i.e. imposing the final state of one configuration to be the initial state of the next.

Digital current mode control: PI current regulator

Digital PI current controller: discrete time synthesis

This approach, that requires the use of exponential matrixes, leads to a general discrete-time state-space model and precisely represents the system dynamic behaviour in the discrete-time domain.

Therefore, in principle, it represents a very good modelling approach for digitally controlled power electronic circuits.

Nevertheless, it is not very commonly used, mainly for two reasons:

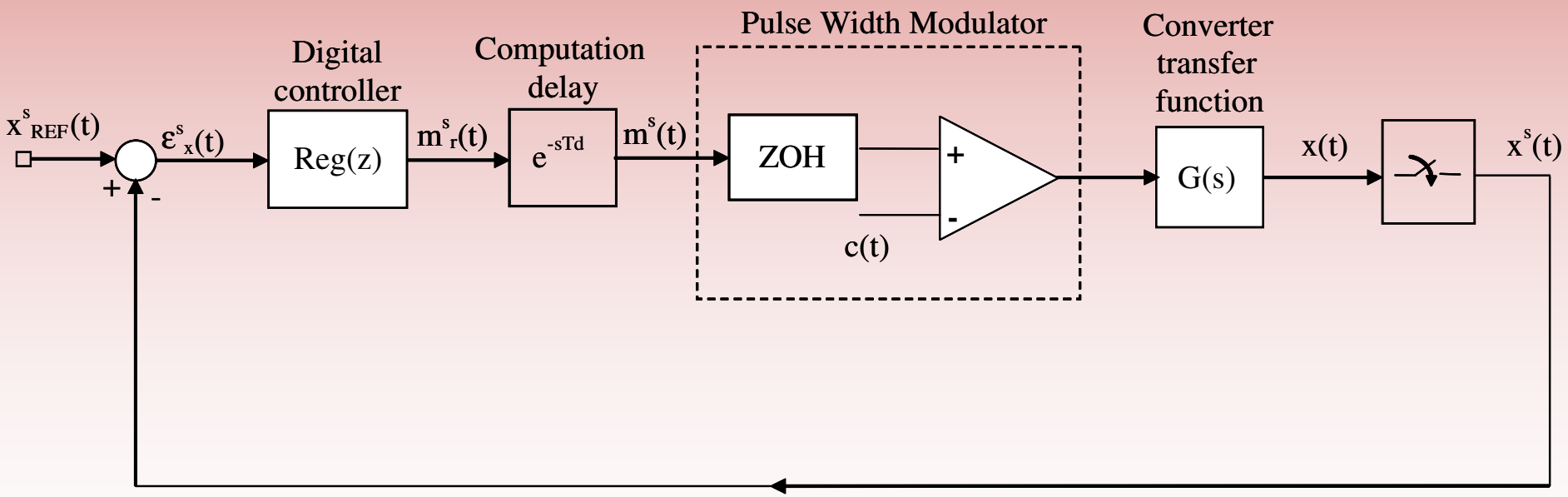
i) the obtained discrete time model depends on the particular type of modulator adopted, as the sequence of state variable integrations, one for each topological state, depends on the modulator mode of operation (leading edge, trailing edge, etc.);

ii) the exponential matrix computation is relatively complex and, therefore, not always practical for the design of power electronic circuit controllers.

Digital current mode control: PI current regulator

Digital PI current controller: discrete time synthesis

A more direct, equivalent, approach to discrete time converter modelling is described in the figure below, where the PWM modulator is represented using the frequency domain model, *PWM(s), G(s),* the converter transfer function, is obtained from the continuous-time converter small-signal model, and $x^s(t)$ is the sampled output variable, that has to be controlled by the digital algorithm.

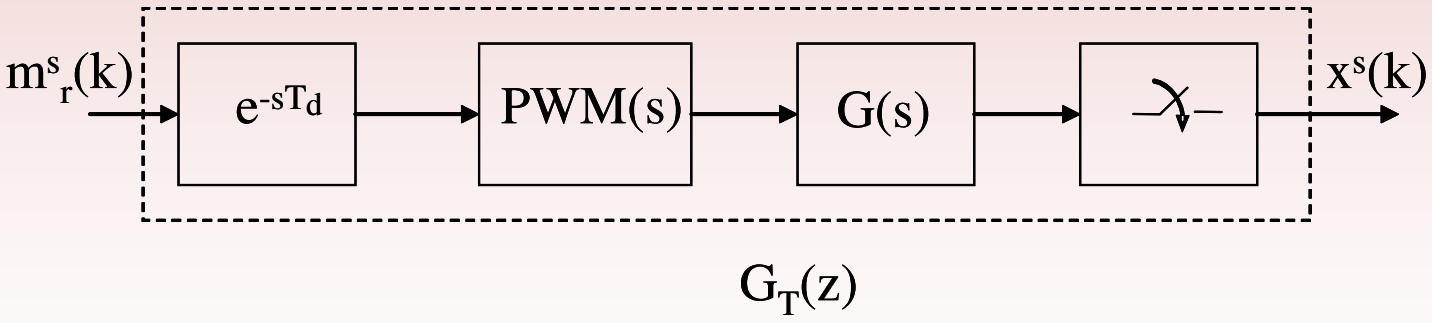


Digital current mode control: PI current regulator

Digital PI current controller: discrete time synthesis

Remember that the Zero Order Hold (ZOH) function that, when cascaded to an ideal sampler, models the conversion from sampled time variables into continuous time variables, is, in our case, internal to the PWM model, and, therefore, does not appear right after the sampler.

Now, if we want to correctly represent the transfer function between the sampled time input variable, m^s_r, and the continuous time output variable of the modulator, a gain equal to T_S has to be added to the modulator transfer function *PWM(s)* as we found out in **Lesson 1.**



Digital current mode control: PI current regulator

Digital PI current controller: discrete time synthesis

This z-domain approach is very powerful: once the Z-transform of the terms included within the dashed box is calculated, it will be capable of correctly quantifying the difference in the converter dynamics determined by the different uniformly sampled modulator implementations (trailing edge, leading edge, triangular carrier modulation, etc..). In addition, it takes into account the exact duty-cycle update instant. The calculation to be performed is the following:

$$G_T(z) = Z[e^{-sT_d} T_s \, PWM(s) \, G(s)]$$

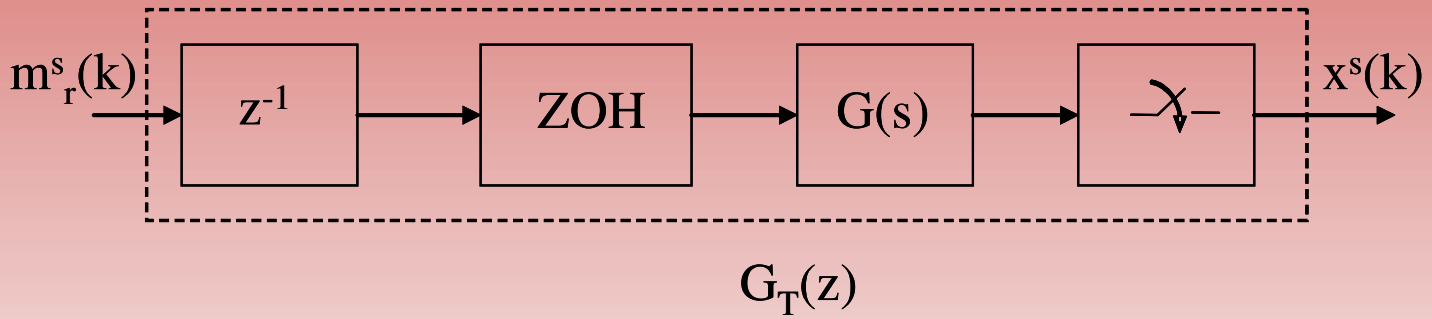
If we are not interested in the exact model, the calculation can be simplified if we observe that:

1. the computation delay is often equal to one modulation period. Therefore, a z^{-1} term can be used to model it instead of the continuous time exponential term.
2. The PWM block transfer function can often be assumed to be equal to that of an ideal ZOH.

Digital current mode control: PI current regulator

Digital PI current controller: discrete time synthesis

Under these simplifying assumptions the block diagram of the continuous time sub system reduces to:



We can rapidly perform the calculation considering a simpler expression for G(s) and the usual expression for the ZOH transfer function, i.e.

$$G(s) = \frac{2 V_{DC} T_S}{s L_S} \qquad \text{and} \qquad H(s) = \frac{1 - e^{-sT_S}}{s}$$

Digital current mode control: PI current regulator

Digital PI current controller: discrete time synthesis

It is immediate to find that, in this case:

$$G_T(z) = z^{-1}Z\left[\frac{1-e^{-sT_S}}{s}\frac{2V_{DC}}{sL_S}\right] =$$

$$= \frac{2V_{DC}}{L_S}z^{-1}(1-z^{-1})Z\left[\frac{1}{s^2}\right] = \frac{2V_{DC}T_S}{L_S}\frac{1}{z(z-1)}$$

It is interesting to observe that, with the particular *G(s)* expression we considered, the exact $G_T(z)$ turns out to be identical to the one above.

Digital current mode control: PI current regulator

Digital PI current controller: discrete time synthesis

Once the transfer function $G_T(z)$ is known, the PI controller design can be performed directly in the discrete time domain. Several methods, e.g. pole allocation, root locus, can be used as the design strategy.

The discrete time synthesis allows us to easily study a different situation, where the controller performance is optimized by implementing a minimum delay regulation loop.

Indeed, in some particular, very demanding cases, even the double update mode of operation for the DPWM excessively penalizes the achievable controller performance.

Much better results can be obtained if the time distance between duty-cycle update and sampling is minimized.

Digital current mode control: PI current regulator

Digital current control with minimum computation delay

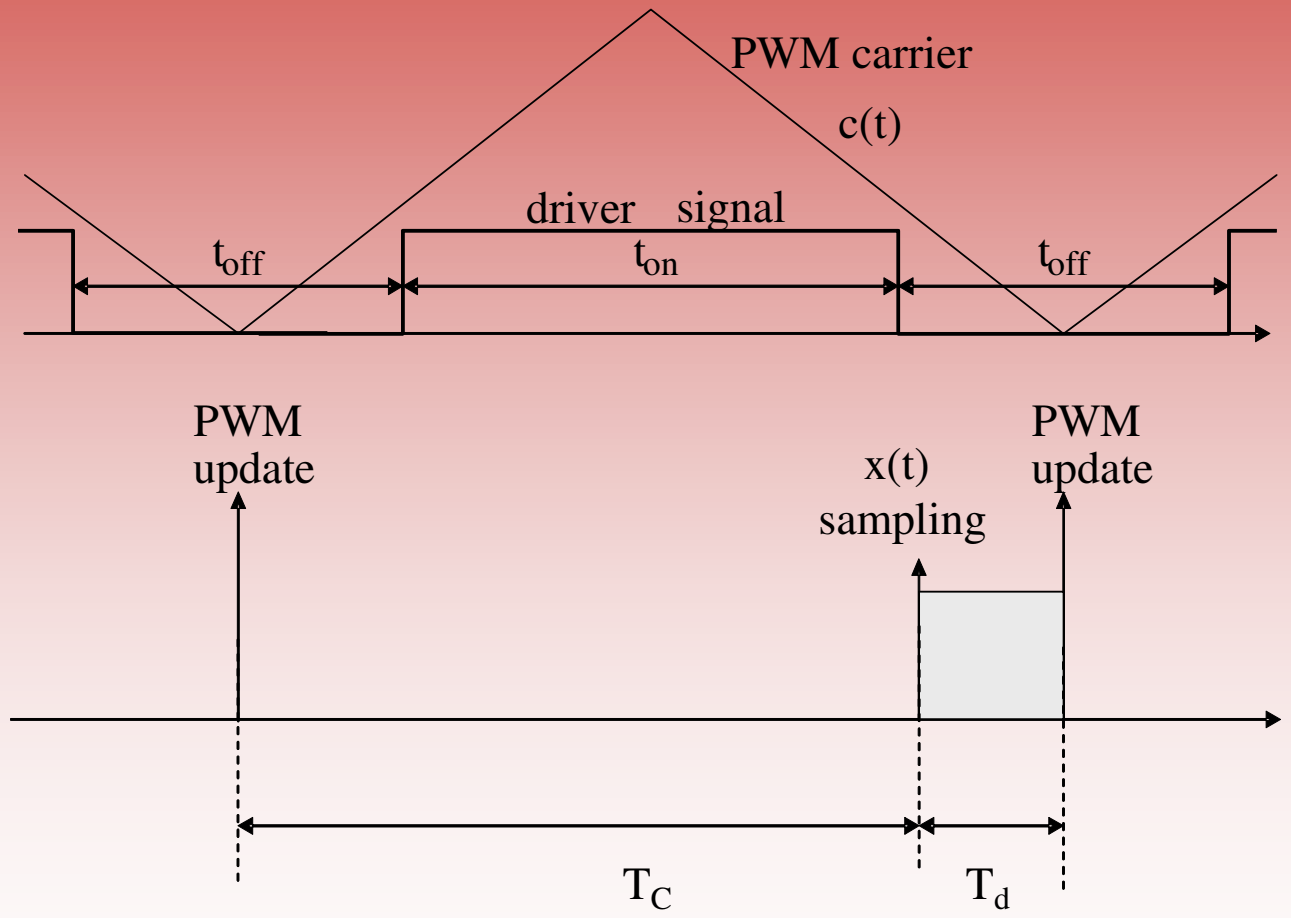
This can be obtained shifting the current sampling instant towards the duty-cycle update instant, leaving just enough time for the ADC to generate the new input sample and to the processor for the control algorithm calculation.

Following this approach and thanks to the continuous increase of microcontrollers' computational power and to the use of DSPs and FPGAs, which are able to compute the control algorithm in smaller and smaller fractions of the switching period, it is possible to reduce the control delay to a little fraction of the sampling period.

The reduced delay, in turn, allows to push the controller bandwidth higher, reducing the gap that separates purely analog and digital control implementations.

Digital current mode control: PI current regulator

Digital current control with minimum computation delay



Digital current mode control: PI current regulator

Digital current control with minimum computation delay

The situation under investigation is depicted in the figure, where T_d is, once again, the time required by AD conversion and calculations.

Time T_C is instead available for other non-critical functions or external control loops.

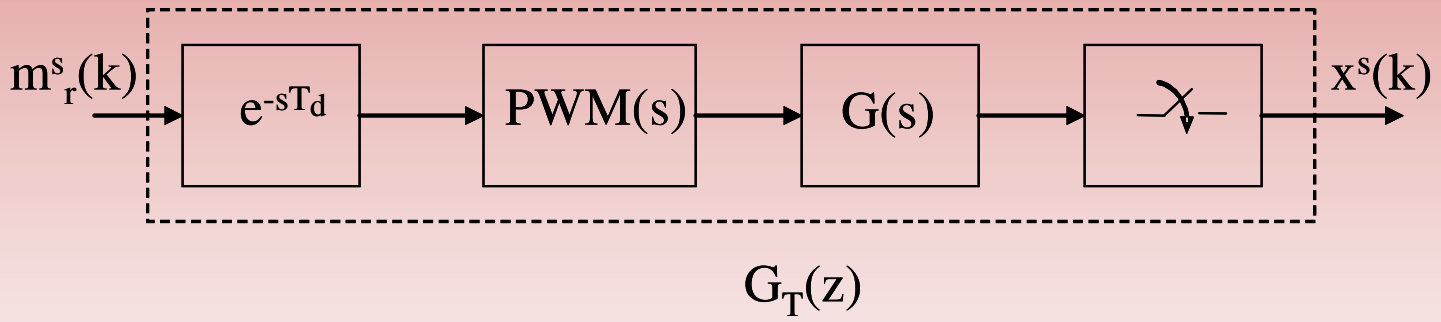
As can be seen, since $T_d << T_S$, being T_S the modulation period, the sampling of the state variable $x(t)$, i.e. in our case of the inductor current $I_O(t)$, is delayed with respect to conventional controller organizations and shifted toward the PWM update instant.

As mentioned above, from the controller's standpoint, this implies a reduction of the feedback loop delay thus allowing a larger regulation bandwidth.

Digital current mode control: PI current regulator

Digital current control with minimum computation delay

In order to quantify the effectiveness of this reduction, an accurate discrete-time model is needed. To this purpose, we can consider again the same block diagram, and replace the PWM block with a Zero-Order-Hold (ZOH), which, as we have seen, represents a very good approximation, especially in the case of triangular carrier waveform.



Now, if the control delay T_d were a sub-multiple of the sampling period T_S, the continuous system could be easily converted into a discrete-time model using conventional Z-transform and considering T_d as the sampling period. In our case, the delay T_d is a generic fraction of sampling period T_S and therefore, modified Z-transform has to be used to correctly model the system.

Digital current mode control: PI current regulator

Digital current control with minimum computation delay

If we define:

$$p = 1 - \frac{T_d}{T_s}$$

we can write the following relation for our block diagram.

$$Z\left[\underbrace{H(s)G(s)}_{G_1(s)}e^{-s(1-p)T_s}\right] = \sum_{k=0}^{\infty} z^{-k} g_1(kT_s - T_d) = Z_m\left[G_1(s)\right] = G_1(z,p)$$

Digital current mode control: PI current regulator

Digital current control with minimum computation delay

Please note that $g_1(t)$ indicates the impulse response of $G_1(s)$ and $G_1(z, p)$ (that we can write as $Z_m[G_1(s)]$) is the modified Z-transform of $G_1(s)$. In the particular case of the Zero Order Hold, H(s) = $(1-e^{-sT_s})/s$ and the previous relation becomes:

$$G_T(z,p) = Z\left[\underbrace{\frac{1-e^{-sT_s}}{s}}_{H(s)} G(s) e^{-s(1-p)T_s}\right] =$$

$$= \frac{z-1}{z} Z\left[\frac{G(s)}{s} e^{-s(1-p)T_s}\right] = \frac{z-1}{z} Z_m\left[\frac{G(s)}{s}\right]$$

Digital current mode control: PI current regulator

Digital current control with minimum computation delay

The modified Z-Transform maintains all the properties of the conventional Z-transform, since it is simply defined as the Z-transform of a delayed signal.

The results of the modified Z-transform application to particular cases of interest are usually available in look-up tables in digital control textbooks.

In our example case, the discrete-time transfer function between the modulating signal *M(z),* input of the DPWM, and the *delayed* inductor current $I_O(z)$ can be written as

$$\frac{I_O(z)}{M(z)} = \frac{2 V_{DC} T_S}{L_S} \cdot \frac{z\,p - (p-1)}{z(z-1)}$$

Note that for p=0, we get the usual expression of slide **32.**

Digital current mode control: PI current regulator

Digital current control with minimum computation delay

In order to quantify the advantages of exactly modelling the delay, i.e. of considering *p >0,* let us take in to account, as a benchmark parameter, the achievable current loop bandwidth.

We assume, for simplicity, that the current regulator is purely proportional and that we want to achieve a given phase margin, for example equal to +50°.

Varying parameter *p*, we now look for the frequency where the transfer function between modulating signal and inverter current shows a -130° phase rotation. That will be the phase rotation of the open loop gain as well, as our controller is purely proportional.

Therefore, we can define that as the achievable current loop bandwidth (BW_i) meaning that a suitable proportional gain exists that makes the transfer function crossover frequency equal to BW_i.

Digital current mode control: PI current regulator

Digital current control with minimum computation delay

Achievable current loop bandwidth (BW_i) versus p

p	0	0.5	0.8
BW_i	$f_s/13.4$	$f_s/9$	$f_s/6.2$

same limit for analog design!

Digital current mode control: PI current regulator

Digital current control with minimum computation delay

Simply by shifting the sampling instant towards the duty-cycle update instant, a significant improvement in the achievable current loop bandwidth can be obtained. For a 20% delay, we practically reach the same condition considered for the analog design example.

It also possible to note that only with $p = 0$ (sampling in the middle of turn-off time) or $p = 0.5$ (sampling in the middle of turn-on time), the sampled current is also the average inductor current, while, for other values of p, some kind of algorithm is needed for the compensation of the current ripple, possibly accounting for dead-time effects as well.

For this reason, the application of the concept here described to current control is fairly complicated, while it can be much more convenient for the control of other system variables, where the switching ripple is smaller.

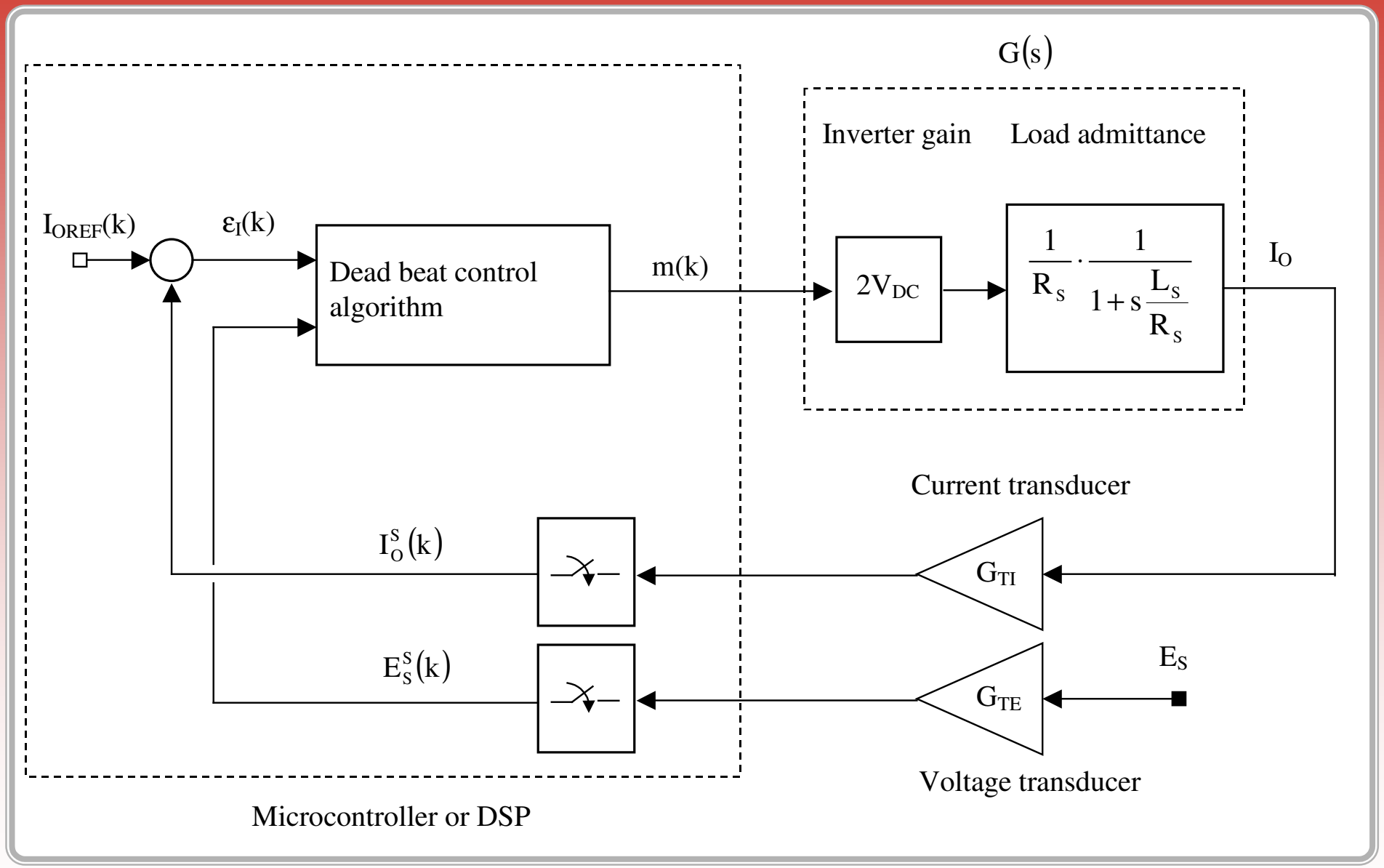
Digital current mode control: predictive controller

We now move to a totally different control approach, describing the predictive, or *dead-beat*, current control implementation. In principle, the dead-beat control strategy we are going to discuss is nothing but a particular application case of discrete time dynamic state feedback and direct pole allocation, and, as such, its formulation for our VSI model can be obtained applying standard digital control theory. However, this *theoretical* approach is not what we are going to follow here. Instead, we will present a different derivation, completely equivalent to the theoretical one, but closer to the physical converter and modulator operation. We will discuss the equivalence of the two approaches later on.

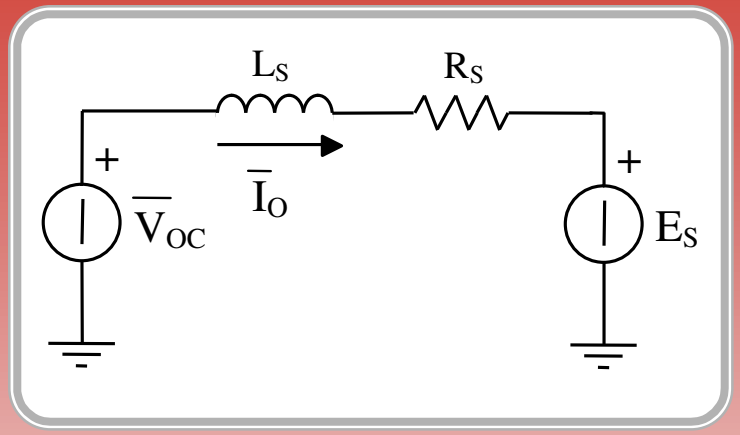
L. Malesani, P. Mattavelli, S. Buso: "Robust Dead-Beat Current Control for PWM Rectifiers and Active Filters", IEEE Transactions on Industry Applications, Vol. 35, No. 3, May/June 1999, pp. 613-620.

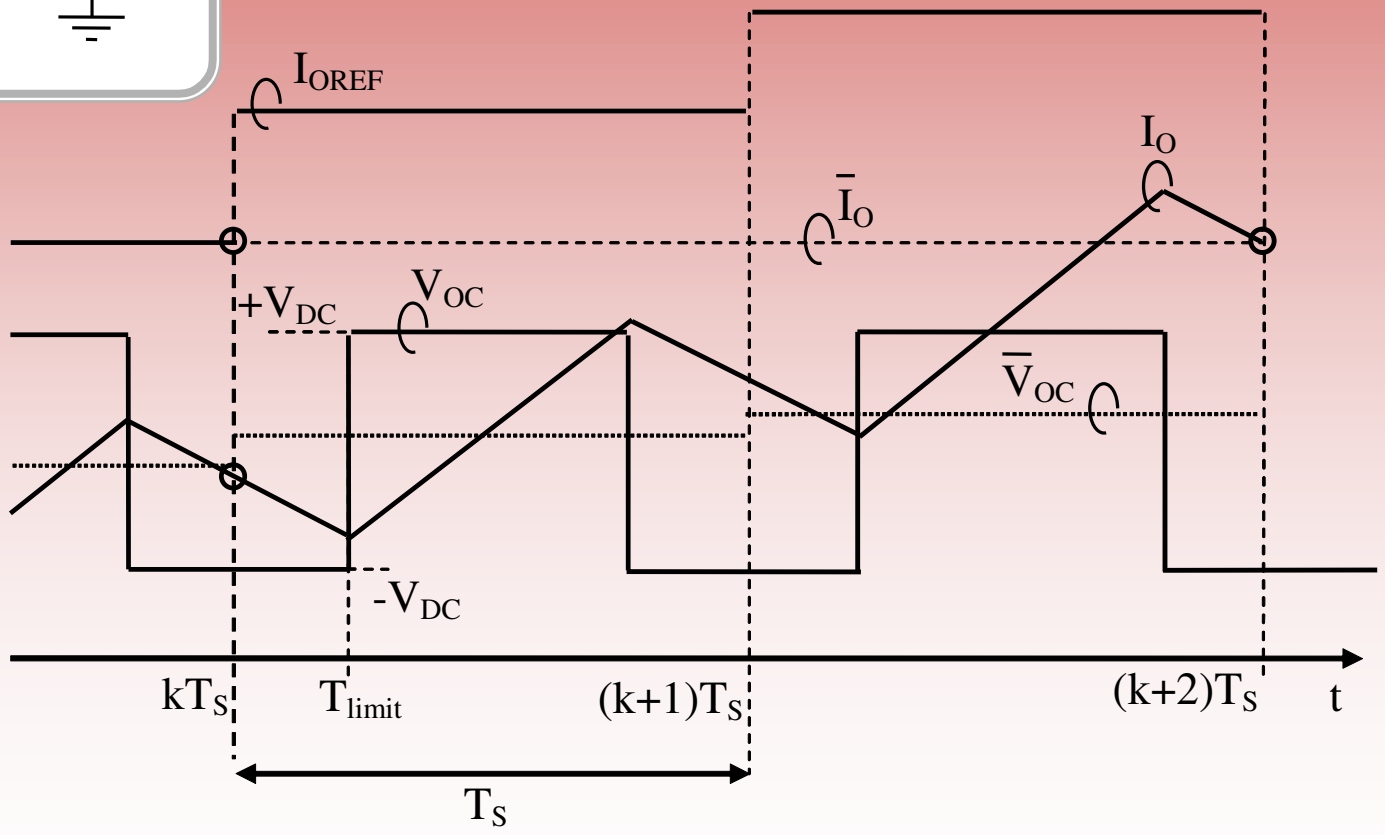
G.H. Bode, P.C. Loh, M.J. Newman, D.G. Holmes "An improved robust predictive current regulation algorithm", IEEE Transactions on Industry Applications, Vol. 41, no. 6, Nov/Dec 2005, pp. 1720-1733.

Digital current mode control: predictive controller



Digital current mode control: predictive controller





Digital current mode control: predictive controller

The reasoning behind the physical approach to predictive current control is quite simple and can be explained referring to an average model of the VSI and its load. At any given control iteration, we want to find the average inverter output voltage, $\bar{V}_{oc}$, that can make the average inductor current, $\bar{I}_o$, reach its reference by the end of the modulation period *following* the one when all the computations are performed. In other words, at instant $k \cdot T_S$ we perform the computation of the $\bar{V}_{oc}$ value that, once generated by the inverter, during the modulation period from $(k+1) \cdot T_S$ to $(k+2) \cdot T_S$, will make the average current equal to its reference at instant $(k+2) \cdot T_S$. The resulting control equation is:

$$\bar{I}_o(k+2) = \bar{I}_o(k+1) + \frac{T_s}{L_s} \cdot \left[\bar{V}_{oc}(k+1) - E_s(k+1)\right] =$$

$$= \bar{I}_o(k) + \frac{T_s}{L_s} \cdot \left[\bar{V}_{oc}(k+1) + \bar{V}_{oc}(k) - E_s(k+1) - E_s(k)\right]$$

Digital current mode control: predictive controller

Assuming now that the phase voltage E_S is a slowly varying signal, as it is often the case, whose bandwidth is much lower than the modulation and sampling frequency, it is possible to consider $E_s(k+1) \cong E_s(k)$, thus obtaining the following dead-beat control equation

$$\overline{V}_{OC}(k+1) = -\overline{V}_{OC}(k) + \frac{L_S}{T_S} \cdot \left[\bar{I}_O(k+2) - \bar{I}_O(k)\right] + 2 \cdot E_S(k)$$

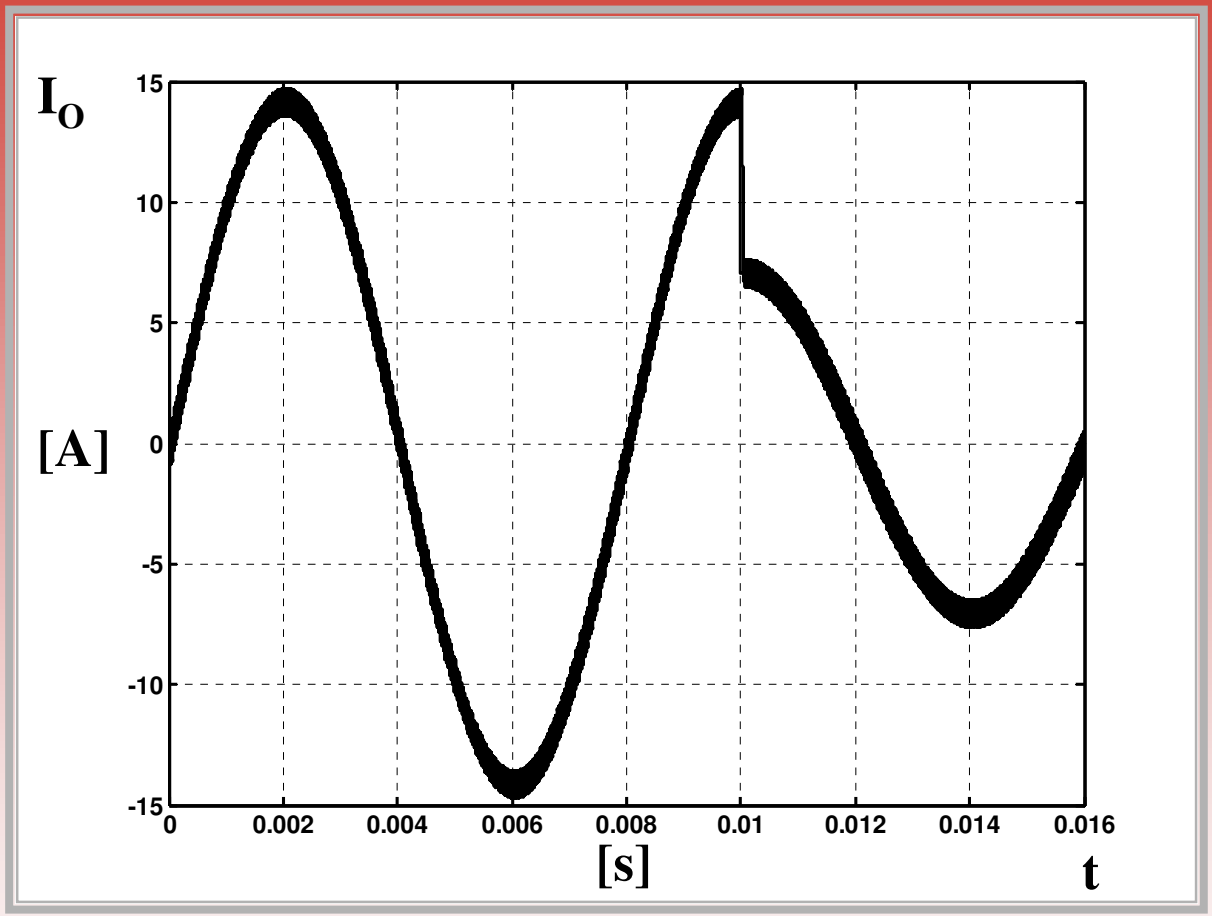
where $\bar{I}_o(k+2)$ can be replaced by $I_{OREF}(k)$, the desired set-point.

Digital current mode control: predictive controller

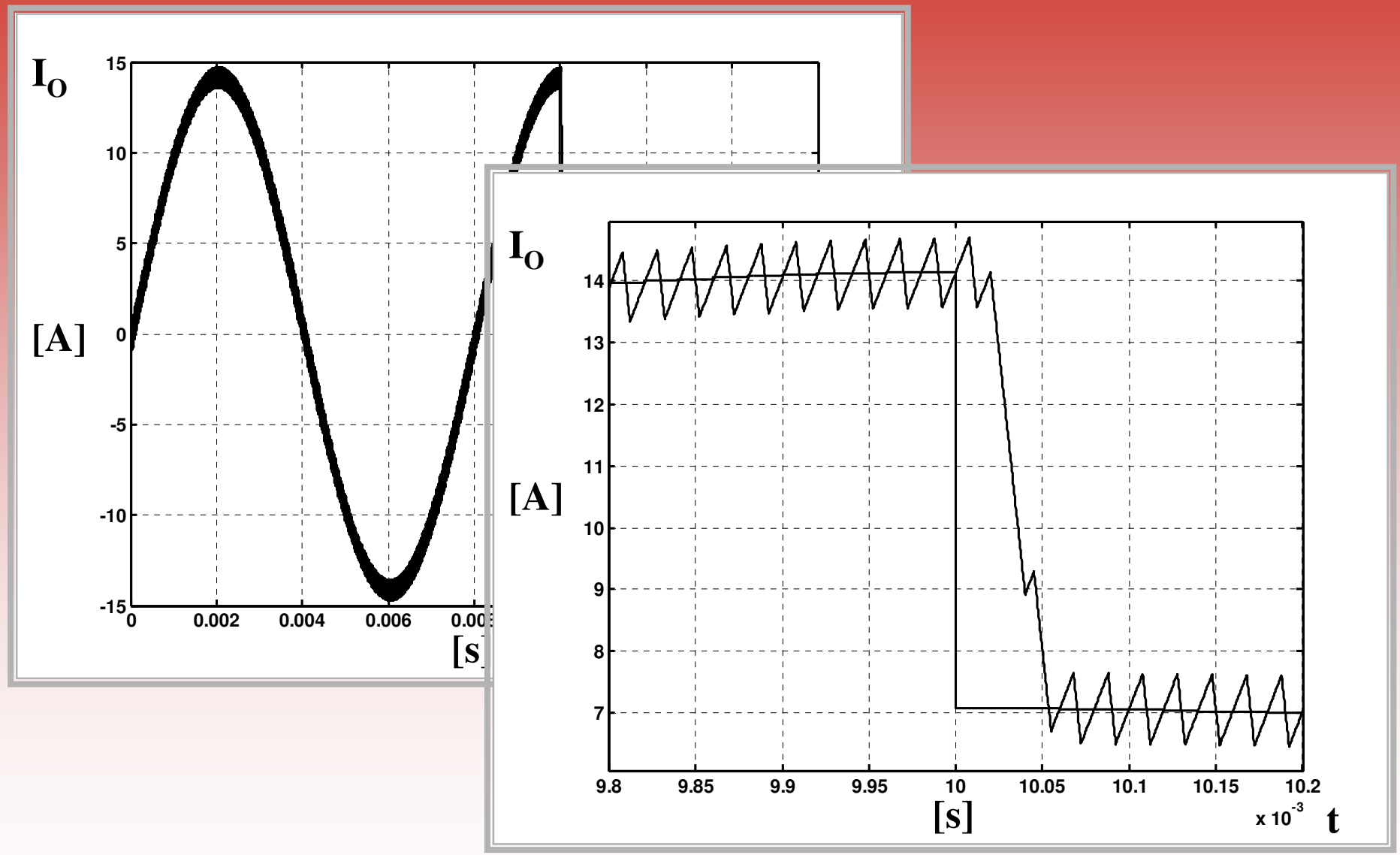
In general, the set-point for the average inverter output voltage it provides us with, will have to be correctly scaled down, so as to fit it to the digital pulse width modulator. The fitting is normally accomplished *normalizing* the output of the controller to the inverter voltage gain. In addition to this, the control equation has to be modified also to properly account for the transducer gains of both current and voltage sensors. It is easy to verify that an equivalent control equation, taking into account the transducer gains and voltage normalization, is the following:

$$m(k+1) = -m(k) + \frac{L_S}{T_S} \cdot \frac{1}{2 \cdot G_{TI} \cdot V_{DC}} \left[I_{OREF\_S}(k) - \bar{I}_{O\_S}(k) \right] + 2 \cdot \frac{1}{2 \cdot G_{TE} \cdot V_{DC}} E_{S\_S}(k)$$

Digital current mode control: predictive controller



Digital current mode control: predictive controller



Digital current mode control: predictive controller

Derivation of the predictive controller through dynamic state feedback

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

Our VSI can be described in the state space that, as we recall from the discussion reported in Lesson 1, can be used to relate average inverter electrical variables. In this case $x = [\bar{I}_O]$ is the state vector, $u = [\overline{V}_{OC}, \overline{E}_S]^T$ is the input vector, $y = [\bar{I}_O]$ is the output variable and the state matrixes are:

$$A = [-R_S/L_S], \quad B = [1/L_S, -1/L_S], \quad C = [1], \quad D = [0, 0]$$

Digital current mode control: predictive controller

It is possible to derive a zero order hold discrete time equivalent of the previous continuous time model considering the following system

$$\begin{cases} x(k+1) = \Phi x(k) + \Gamma u(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

where, by definition, $\Phi = e^{A \cdot T_S}$ and $\Gamma = (\Phi - I) \cdot A^{-1} \cdot B$

Computation of Φ and Γ yields:

$$\Phi = e^{A \cdot T_S} = e^{-\frac{R_S}{L_S} \cdot T_S} \xrightarrow{R_S \to 0} 1$$

$$\Gamma = \begin{bmatrix} -\dfrac{e^{-\frac{R_S}{L_S} \cdot T_S} - 1}{R_S} & \dfrac{e^{-\frac{R_S}{L_S} \cdot T_S} - 1}{R_S} \end{bmatrix} \xrightarrow{R_S \to 0} \begin{bmatrix} \dfrac{T_S}{L_S} & -\dfrac{T_S}{L_S} \end{bmatrix}$$

Digital current mode control: predictive controller

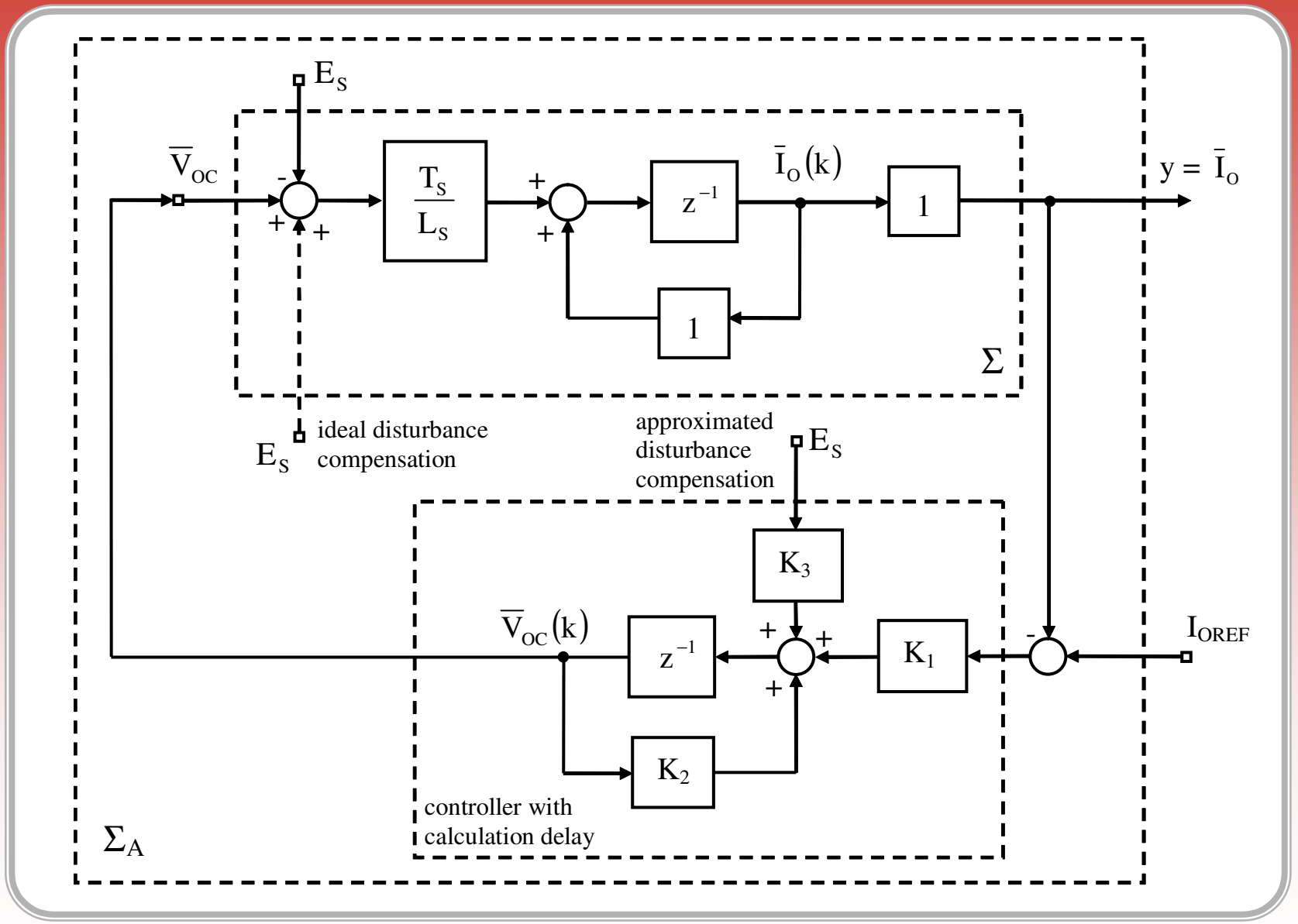
We can derive the predictive controller as a particular case of state feedback and pole placement. In order to show that, we may re-write the state equations explicitly. We get the following result:

$$\Sigma: \begin{cases} \bar{I}_O(k+1) = \bar{I}_O(k) + \dfrac{T_S}{L_S} \cdot \overline{V}_{OC}(k) - \dfrac{T_S}{L_S} \cdot \overline{E}_S(k) \\ y(k) = \bar{I}_O(k) \end{cases}$$

We can then represent the controller by means of the following state equation:

$$\overline{V}_{OC}(k+1) = K_2 \cdot \overline{V}_{OC}(k) + K_1 \cdot \left[I_{OREF}(k) - \bar{I}_O(k) \right]$$

Digital current mode control: predictive controller



Digital current mode control: predictive controller

The interconnection of Σ and the controller feedback generates a new, augmented, dynamic system, indicated by Σ_A. This is described by the following equations:

$$\Sigma_A : \begin{cases} \bar{I}_O(k+1) = \bar{I}_O(k) + \dfrac{T_S}{L_S} \cdot \overline{V}_{OC}(k) - \dfrac{T_S}{L_S} \cdot \overline{E}_S(k) \\ \overline{V}_{OC}(k+1) = K_2 \cdot \overline{V}_{OC}(k) + K_1 \cdot \left[I_{OREF}(k) - \bar{I}_O(k) \right] + K_3 \cdot \overline{E}_S(k) \\ y(k) = [1 \quad 0] \cdot \begin{bmatrix} \bar{I}_O(k) \\ \overline{V}_{OC}(k) \end{bmatrix} \end{cases}$$

that correspond to the state vector augmentation to $x_A = [\ \bar{I}_O\ \overline{V}_{OC}\ ]^T$, to the new input vector $u_A = [\ \overline{E}_S\ I_{OREF}\ ]^T$ and to the *approximated* compensation of the exogenous disturbance, governed by gain K_3.

Digital current mode control: predictive controller

The corresponding state matrixes are the following:

$$\Phi_A = \begin{bmatrix} 1 & \frac{T_S}{L_S} \\ -K_1 & K_2 \end{bmatrix} \quad \Gamma_A = [\Gamma_{A1} \,\vdots\, \Gamma_{A2}] = \left[\begin{array}{c:c} -\frac{T_S}{L_S} & 0 \\ K_3 & K_1 \end{array}\right]$$

$$C_A = [1 \quad 0], \; D_A = [0 \quad 0]$$

Digital current mode control: predictive controller

It is possible to determine parameters K_1, K_2 and K_3 to get the desired pole allocation and disturbance compensation. It is easy to verify that choosing:

$$K_1 = \frac{L_S}{T_S}, \qquad K_2 = -1$$

the eigenvalues of Σ_A move to the origin of the complex plane. As is well known, this is a sufficient condition to achieve a dead-beat closed loop response from the controlled system. Alternatively, the position of poles on the complex plane can be chosen to achieve a different closed loop behaviour, for example one equivalent to that of a continuous time, first order, stable system, characterized by any desired time constant. Indeed, with the direct discrete time design of the regulator, the designer has, in principle, complete freedom in choosing the preferred pole allocation.

Digital current mode control: predictive controller

Applying standard state feedback theorems and after simple calculations we find:

$$\frac{\bar{I}_O}{I_{OREF}}(z) = C_A \cdot (zI - \Phi_A)^{-1} \cdot \Gamma_{A2} = \frac{1}{z^2}$$

which corresponds, as expected, to a dynamic response equivalent to a pure two modulation period delay. Similarly, we can compute the closed loop transfer function from the disturbance to the output. We find:

$$\frac{\bar{I}_O}{E_S}(z) = C_A \cdot (zI - \Phi_A)^{-1} \cdot \Gamma_{A1} = \frac{1}{z^2} \cdot \frac{T_S}{L_S} \cdot (-z - 1 + K_3)$$

Digital current mode control: predictive controller

As can be seen, there is no value of K_3 that can guarantee a zero transfer function from disturbance to output. This is due to the fact that the compensation term of the controller equation is one step delayed with respect to the control output and, as such, is only approximated. In these conditions, the best we can do is to minimize the transfer function between disturbance and output. It is easy to verify that the choice $K_3 = 2$ achieves this minimization. Rewriting the equation in the time domain and imposing $K_3 = 2$ we find:

$$\bar{I}_O(k) = \frac{T_S}{L_S} \cdot \left[-E_S(k-1) + E_S(k-2) \right]$$

that, under the assumption of slowly varying E_S, guarantees the minimum disturbance effect of the output. Having determined the controller parameters K_1, K_2 and K_3, we are now ready to explicitly write the control equation, that turns out to be:

.

$$\overline{V}_{OC}(k+1) = -\overline{V}_{OC}(k) + \frac{L_S}{T_S} \cdot \left[I_{OREF}(k) - \bar{I}_O(k) \right] + 2 \cdot \overline{E}_S(k)$$

Digital current mode control: predictive controller

A simple improvement of the predictive controller is obtained deriving an estimation equation, that allows to save the measurement of the phase voltage E_S. As in the control equation's case, the estimation equation can be derived by simple physical considerations. Indeed, re-writing the control equation one step backward we get

$$\bar{I}_O(k) - \bar{I}_O(k-1) = \frac{T_S}{L_S} \cdot \left[\overline{V}_{OC}(k-1) - E_S(k-1)\right]$$

from which we can extract an estimation of $E_S(k\text{-}1)$. Simple manipulations of (3.2.17) yield

$$\hat{E}_S(k-1) = \overline{V}_{OC}(k-1) - \frac{L_S}{T_S} \cdot \left[\bar{I}_O(k) - \bar{I}_O(k-1)\right]$$

It is typically possible to improve the quality of the estimation by using some form of interpolation or filtering, that can remove possible estimator instabilities.

Digital current mode control: predictive controller

Robustness of the predictive controller

Considering the dead beat control equation, we see that several parameters contribute to the definition of the algorithm coefficients, each of them being a potential source of mismatch. To give an example of the analysis procedure we can apply to estimate the sensitivity of the controller to the mismatch, we begin by referring, for simplicity, to the following simplified equation, where the only parameter we need to take into account is inductor L_S.

$$\overline{V}_{OC}(k+1) = -\overline{V}_{OC}(k) + \frac{L_S \pm \Delta L_S}{T_S} \cdot \left[\bar{I}_O(k+2) - \bar{I}_O(k)\right] + 2 \cdot E_S(k)$$

Note that parameter L_S has been replaced by $L_S \pm \Delta L_S$, thus putting into evidence the possible presence of an error, ΔL_S, implicitly defined as a **positive** quantity.

Digital current mode control: predictive controller

The analysis of the impact of ΔL_S on the system's stability requires the computation of the system's eigenvalues. Referring to the procedure outlined in before, we can immediately find the state matrix corresponding to the previous equation and the usual control equation. This turns out to be:.

$$\Phi'_A = \begin{bmatrix} 1 & \frac{T_S}{L_S} \\ -\frac{L_S \pm \Delta L_S}{T_S} & -1 \end{bmatrix}$$

Φ'_A

It is now immediate to find the eigenvalues of the above matrix. These are given by the following expression:

$$\lambda_{1,2} = \sqrt{\pm \frac{\Delta L_S}{L_S}} \quad \Rightarrow \quad |\lambda_{1,2}| = \sqrt{\frac{\Delta L_S}{L_S}}$$

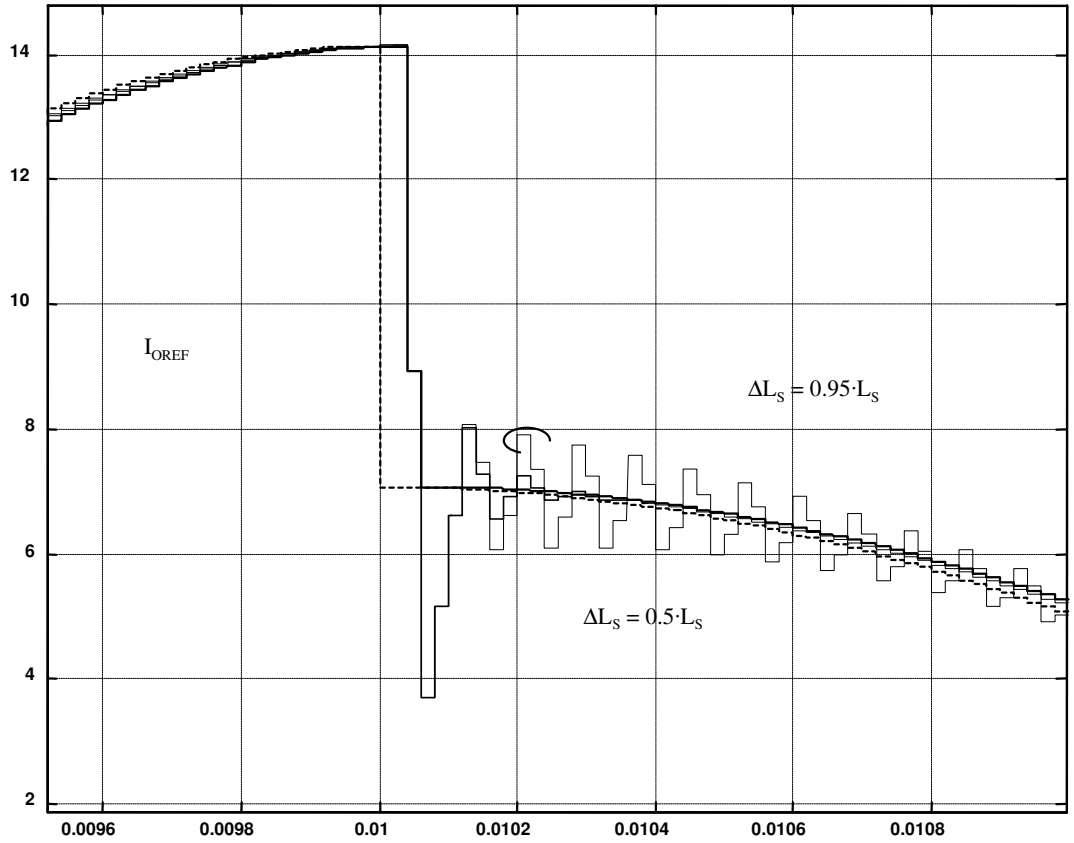
Digital current mode control: predictive controller

From the above result we see that the magnitude of the closed loop system's eigenvalues is limited to the square root of the relative error on L_S. This means that, unless a higher than 100% error is made on the estimation of L_S or, equivalently, unless a 100% variation of L_S takes place, due to changes in the operating conditions, the predictive controller will keep the system stable.

Please note that, interestingly, this result is *independent* of the sampling frequency.

Of course, even if instability requires bigger than unity eigenvalues, the good reference tracking properties of the predictive controller are likely to get lost, even for smaller than unity values of the relative error.

Digital current mode control: predictive controller



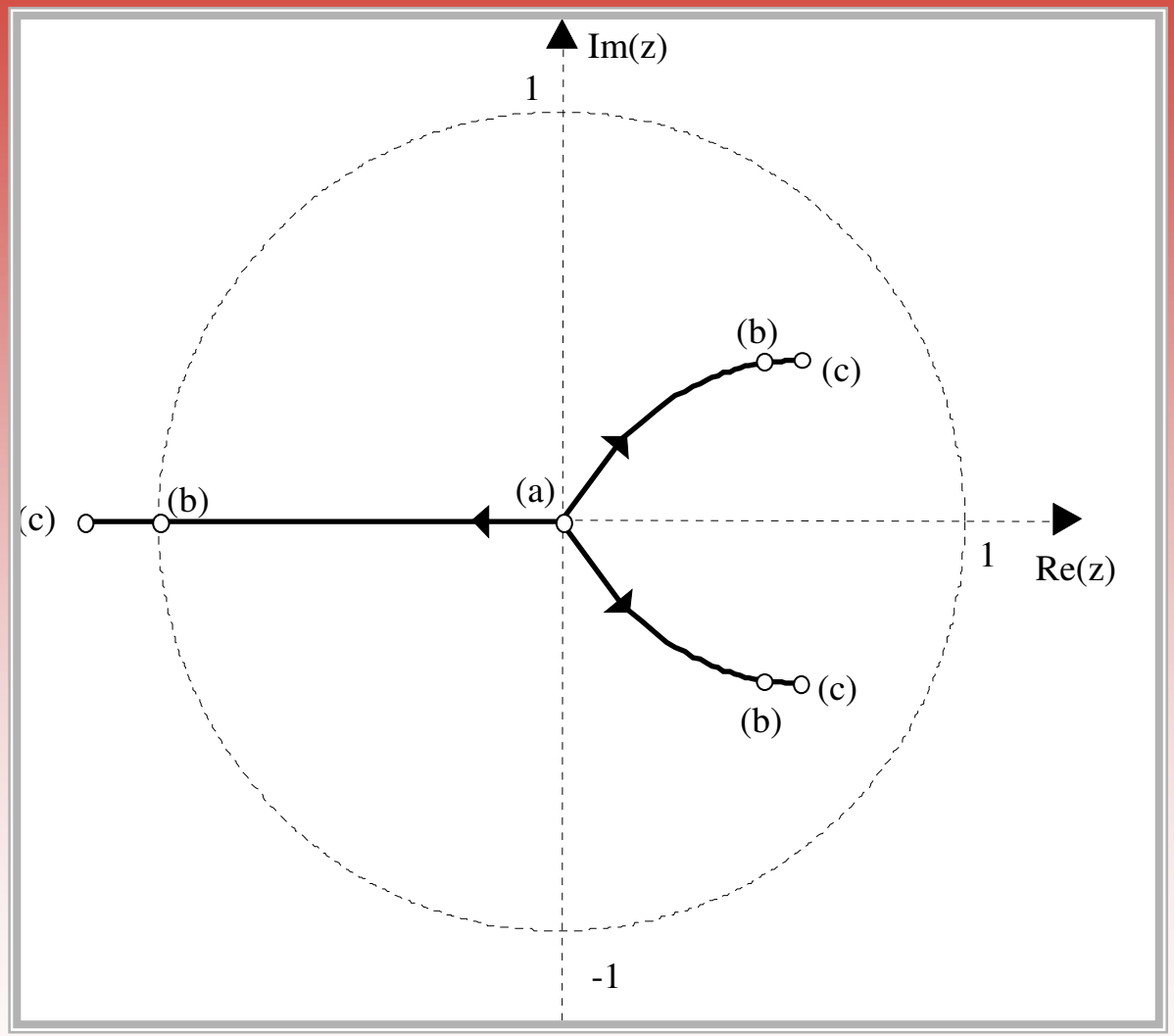
Digital current mode control: predictive controller

Robustness of the predictive controller

As it might be expected, the robustness of the predictive controller to mismatches gets worse if the estimation of the phase voltage is used instead of its measurement. The analytical investigation of this case is a little more involved than the previous one, but still manageable with pencil and paper calculations. The procedure consists in writing the system, controller and estimator equations, either solving them using Z-transform to find the reference to output transfer function, or, equivalently, arranging them to get the state matrix, and, finally, examining the characteristic polynomial of the system. Following this procedure, we get:

$$\lambda(z) = z^3 \pm 3 \cdot \frac{\Delta L_s}{L_s} \cdot z \pm 2 \cdot \frac{\Delta L_s}{L_s}$$

Digital current mode control: predictive controller



Plot of the closed loop system eigenvalues as functions of the parameter L_S mismatch.

a) $\Delta L_S = 0$.

b) $\Delta L_S = 0.2 \cdot L_S$.

c) $\Delta L_S = 0.3 \cdot L_S$.

Digital current mode control: predictive controller

Robustness of the predictive controller

Converter dead-times are another non-ideal characteristic of the VSI that is not taken into account by the model the predictive controller is based on. In a certain sense, their presence can be considered a particular case of model mismatch. We know from previous lessons that the presence of dead-times implies a systematic error on the average voltage generated by the inverter. The error has an amplitude that depends directly on the ratio between dead-time duration and modulation period and a sign that depends on the load current sign. As we did before, we can model the dead times' effect as a square-wave disturbance having a relatively small amplitude (roughly a few percent of the dc link voltage) and opposite phase with respect to the load current. We can consider this disturbance as an undesired component that is summed, at the system input, to the average voltage requested by the current control algorithm.

Digital current mode control: predictive controller

Robustness of the predictive controller

The disturbance should be, at least partially, rejected by the current controller. The effectiveness of the input disturbance rejection capability depends on the low frequency gain the controller is able to determine for the closed loop system. And here is where the dead-beat controller shows another weak point. We have seen how the dead-beat action tends to get from the closed loop plant a dynamic response that is close to a pure delay. Unfortunately, this implies a very poor rejection capability for any input disturbance. To clarify this point we can again compute the closed loop transfer function from the exogenous disturbance $\bar{E}_S$ to the output $\bar{I}_O$. Indeed, this is the transfer function experienced by the dead-time induced voltage disturbance. Simple calculations yield:

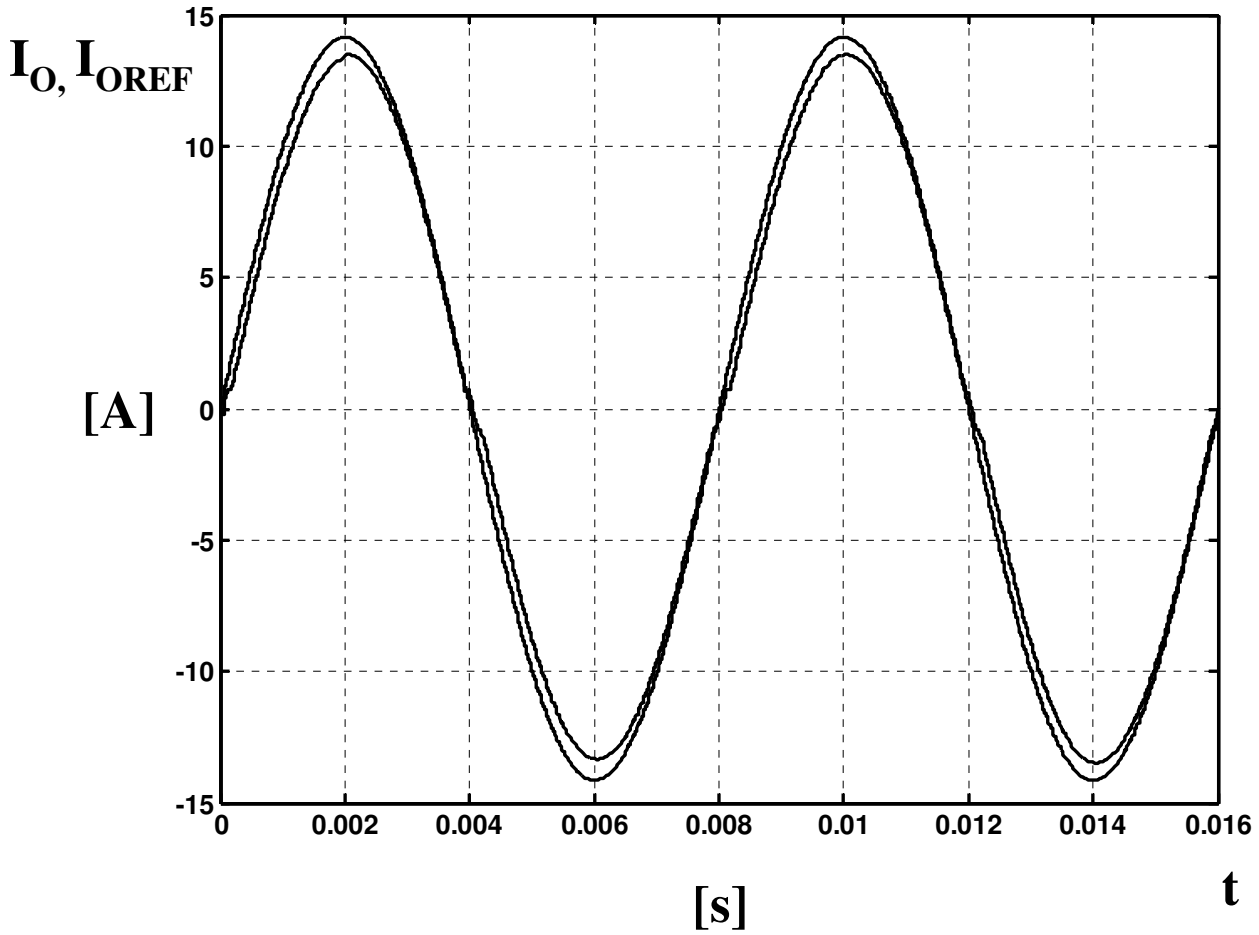
$$\frac{\bar{I}_O}{\bar{E}_S}(z) = \frac{1}{z^2} \cdot \frac{T_S}{L_S} \cdot (z+1)$$

Digital current mode control: predictive controller

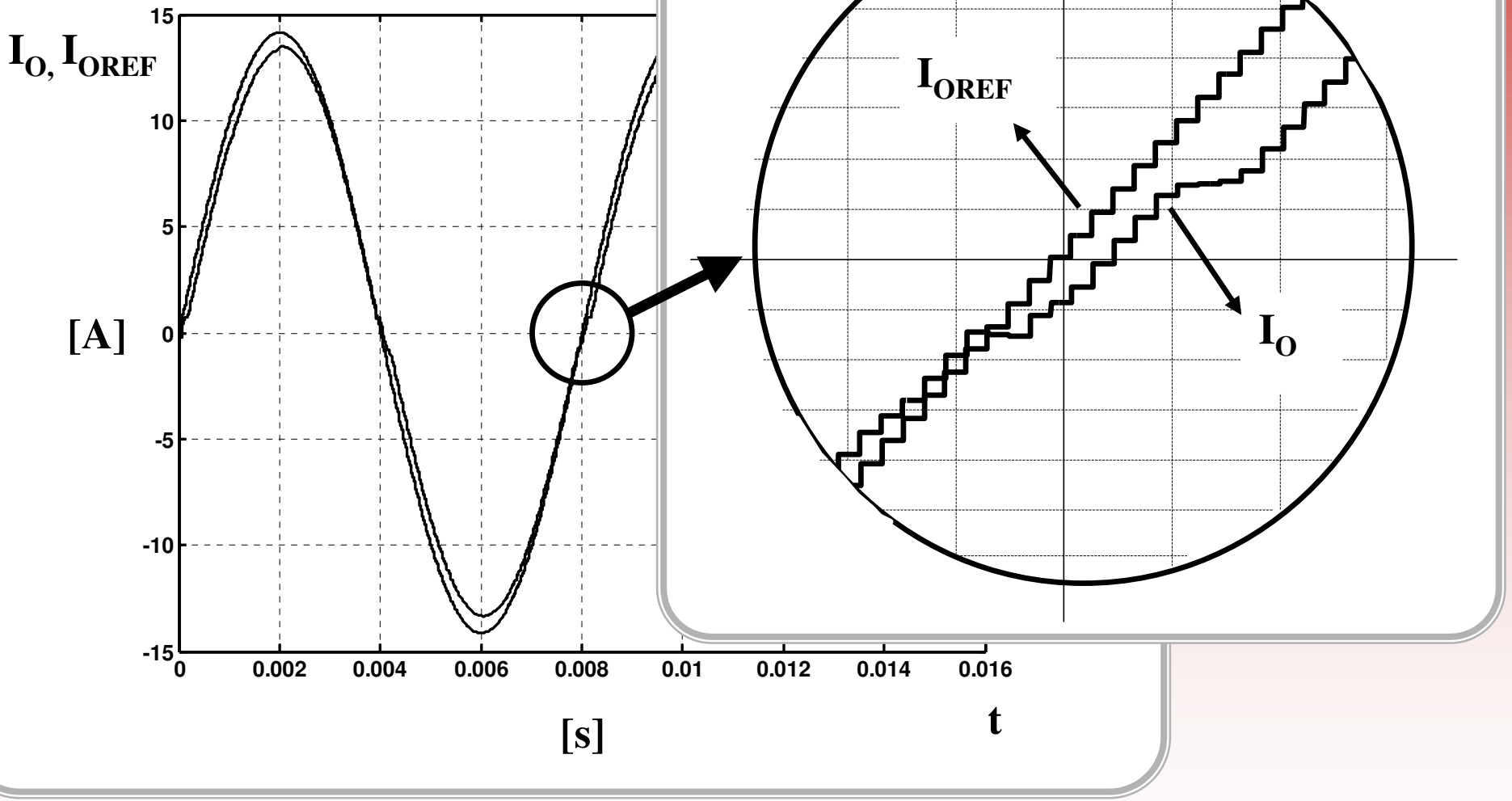
Robustness of the predictive controller

In terms of disturbance rejection this result is rather disappointing. Plotting the frequency response we find that it is practically flat from zero up to the Nyquist frequency, i.e. there is no rejection of the average inverter voltage disturbance. Consequently, we cannot expect the deadbeat controller to compensate the dead times' effect. This means that, unless some external, additional compensation strategy is adopted, a certain amount of current distortion is likely to be encountered.

Digital current mode control: predictive controller



Digital current mode control: predictive controller



Digital current mode control: predictive controller

Dead time compensation strategies

The dead-beat controller requires some form of dead time compensation. Compensation methods can be divided into: *i)* closed loop or on-line and *ii)* open loop or off-line.

The best performance is offered by closed loop dead time compensation, that requires, however, the measurement of the actual inverter average output voltage. Its comparison with the voltage set-point provided to the modulator gives sign and amplitude of the dead-time induced average voltage error, that can therefore be compensated with minimum delay, simply by summing to the set-point for the following modulation period the opposite of the measured error.

The need for measuring the typically high output inverter voltage requires the use of particular care. The estimation of the inverter average output voltage is normally done by measuring the duration of the voltage-high and voltage-low parts of the modulation period, i.e. by computing the actual, effective output duty-cycle.

Digital current mode control: predictive controller

Dead time compensation strategies

However, much more often, off line compensation strategies are used. These offer a lower quality compensation, but can be completely embedded in the modulation routine programmed in the microcontroller (or DSP), requiring no measure.

The off line compensation of dead-times is based on a worst case estimation of the dead-time duration and on the knowledge of the sign of the output current, that is normally inferred from the reference signal (not from the measured output current, to avoid any complication due to the high frequency ripple). Given both of these data, it is possible to add to the output voltage set-point a compensation term that balances the dead-time induced error.

The method normally requires some tuning, in order to avoid under or over compensation effects. The results are normally quite satisfactory, unless a very high precision is required by the application, allowing to eliminate the amplitude error and to strongly attenuate the crossover distortion phenomenon.