Elements of Power Electronics
PART II: Digital control

Fabrice Frébel (fabrice.frebel@uliege.be)

September 15th, 2025

PART II: Digital control

- Chapter 1: Continuous-Time Averaged Modeling of DC-DC Converters
- Chapter 2: The Digital Control Loop
- Hands-on: The complete design process
- Current Mode Control

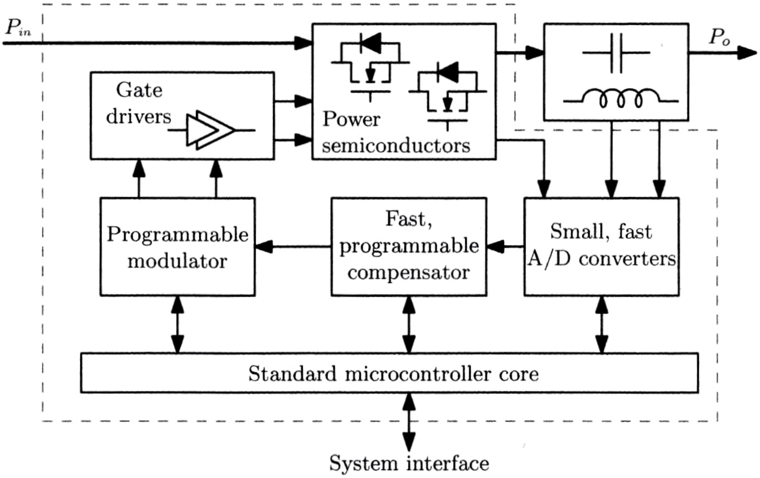
PART II is based on the reference book [1] with same chapter numbering.

Chapter 1: Continuous-Time Averaged Modeling of DC-DC Converters

- Digitally controlled switched-mode converters
- Converters transfer function
  - Solving the time variance problem: averaging
  - Solving the non-linearity problem: converter linearization
  - Transfer function: buck converter example
  - Another example: buck-boost converter
  - Averaged small-signal models of basic converters
  - Boost transfer function and right half-plane zero (RHPZ)
  - State-space averaging
- The pulse width modulator
- Closed loop system
  - Definition of the loop gain
  - Loop including external perturbation
  - Closed loop gain
  - Relation between phase margin and stability
- Analog control loop design procedure

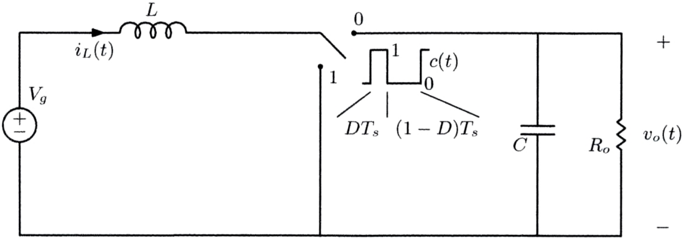
Digitally controlled switched-mode converters

Excerpt of [1]:


P_{in}
P_o
Gate drivers
Power semiconductors
Programmable modulator
Fast, programmable compensator
Small, fast A/D converters
Standard microcontroller core
System interface

Pulse width modulated converter

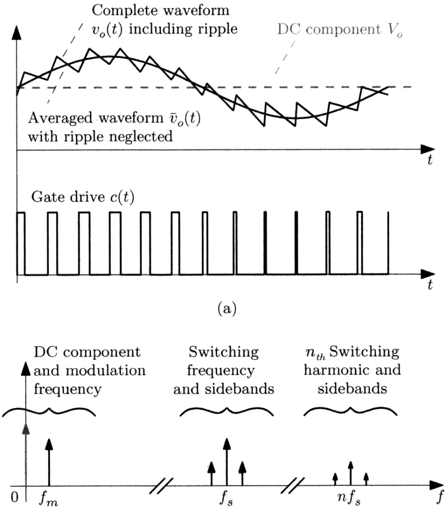
Excerpt of [1]:



- We have tools to study LTI (Linear Time Invariant) systems but,
- pulse width modulated converter are non-linear (M(D) is often not linear) and time variant (switching).

Solving the time variance problem: averaging

Excerpt of [1]:



(a)

(b)

To solve the time variance, we apply the moving average operator with period T:

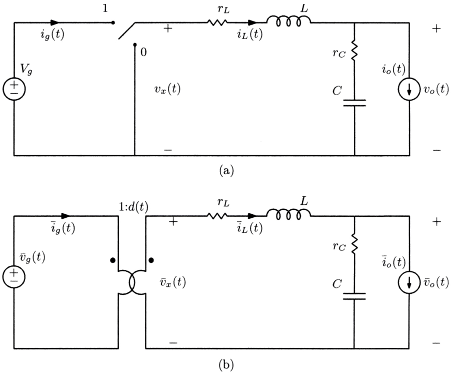
$$\langle x(t)\rangle_T \triangleq \frac{1}{T}\int_{t-T/2}^{t+T/2} x(\tau)d\tau.$$

The goal is to obtain a model of averaged variables over a switching period, this yields for $v_o(t)$:

$$\boxed{\bar{v}_o(t) \triangleq \langle v_o(t)\rangle_{T_s}.}$$

Converter averaging: buck converter example

Excerpt of [1]:



(a)

(b)

All variables that are "switched" have to be averaged. For the buck converter of figure (a), 2 variables have to be considered:

$$
\begin{aligned}
\bar{v}_x(t) &\approx d(t)\bar{v}_g(t),\\
\bar{i}_g(t) &\approx d(t)\bar{i}_L(t).
\end{aligned}
$$

The resulting averaged model is shown on figure (b).

Note: $d(t)$ is not "switched" and must not be averaged.

Solving the non-linearity problem: converter linearization

Let us first define circuit variables as follows:

$$\bar{x}(t) = X + \hat{\bar{x}}(t)$$

$\bar{x}(t)$: averaged value of variable x

X: dc component of variable x (= operating point)

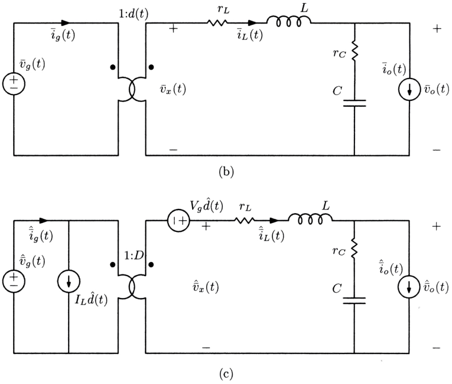
$\hat{\bar{x}}(t)$: small-signal value of variable x around X

Products of variables (source of non linearities) can be linearized:

$$\begin{aligned}\bar{x}(t)\bar{y}(t) =& (X + \hat{\bar{x}}(t))(Y + \hat{\bar{y}}(t)) \\ =& XY + X\hat{\bar{y}}(t) + \hat{\bar{x}}(t)Y + \hat{\bar{x}}(t)\hat{\bar{y}}(t) \\ \approx& XY + X\hat{\bar{y}}(t) + \hat{\bar{x}}(t)Y.\end{aligned}$$

Converter linearization: buck converter example

Excerpt of [1]:



The buck converter of figure (b) can then be linearized:

$$\begin{aligned}\bar{v}_x(t) &= V_x + \hat{\bar{v}}_x(t)\\ &= (D + \hat{d}(t))(V_g + \hat{\bar{v}}_g(t))\\ &\approx DV_g + D\hat{\bar{v}}_g(t) + \hat{d}(t)V_g.\end{aligned}$$

Therefore,

$$\hat{\bar{v}}_x(t) \approx D\hat{\bar{v}}_g(t) + \hat{d}(t)V_g.$$

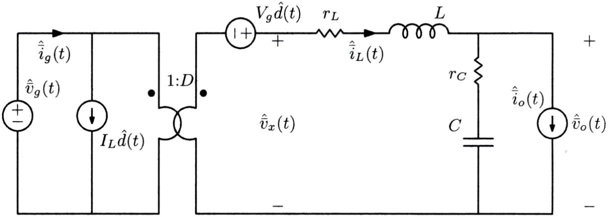
In the same way,

$$\hat{\bar{i}}_g(t) \approx D\hat{\bar{i}}_L(t) + \hat{d}(t)I_L.$$

The result of the linearization is shown on figure (c).

Transfer function: buck converter example

Excerpt of [1]:



(c)

The small-signal transfer function of the buck converter is:

$$G_{vd}(s) \triangleq \left.\frac{\hat{v}_o(s)}{\hat{d}(s)}\right|_{\hat{v}_g=0,\hat{i}_o=0} = V_g \frac{1 + sr_C C}{1 + s(r_C + r_L)C + s^2 LC}$$

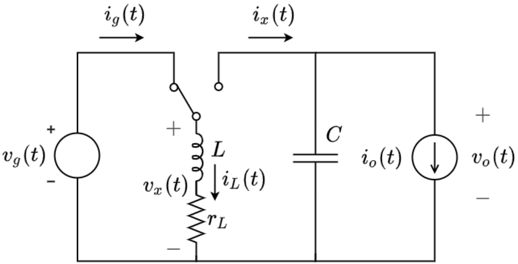
$$= G_{vd0} \frac{1 + \frac{s}{\omega_{ESR}}}{1 + \frac{s}{Q\omega_0} + \frac{s^2}{\omega_0^2}}.$$

Transfer function: buck converter example

In the previous equation, the constants are defined as follows:

$$
\begin{aligned}
G_{vd0} &\triangleq V_g, \\
\omega_{ESR} &\triangleq \frac{1}{r_C C}, \\
\omega_0 &\triangleq \frac{1}{\sqrt{LC}}, \\
Q &\triangleq \frac{1}{r_C + r_L}\sqrt{\frac{L}{C}}.
\end{aligned}
$$

Another example: buck-boost converter



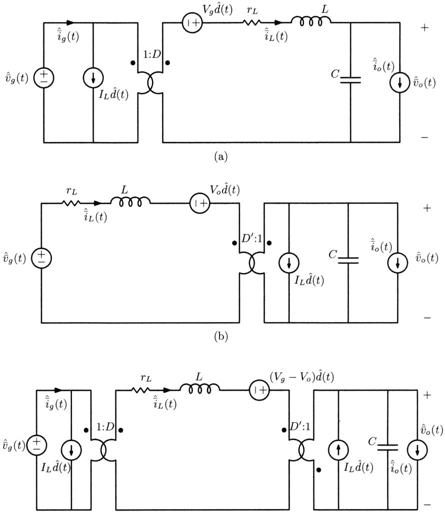
To obtain the small-signal model of the buck-boost:

- write averaged values of $\bar{v}_x(t)$, $\bar{i}_x(t)$ and $\bar{i}_g(t)$,
- linearize the averaged values to get $\hat{\bar{v}}_x(t)$, $\hat{\bar{i}}_x(t)$ and $\hat{\bar{i}}_g(t)$,
- draw the circuit that matches these linearized relationship.

Write the equations and draw the model

Summary: averaged small-signal models of basic converters

Excerpt of [1]:



(a)

(b)

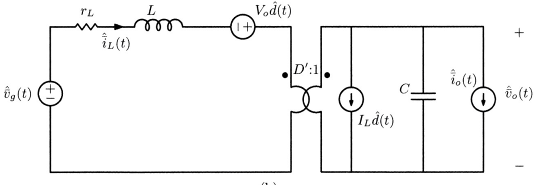
(c)

(a) Buck

(b) Boost

(c) Buck-Boost

Boost transfer function and right half-plane zero (RHPZ)



(b)

The small-signal transfer function of the boost converter is also obtained by solving the above circuit, but two controlled sources are involved in this case (using superposition is a fast way to find the solution):

$$G_{vd}(s) \triangleq \left.\frac{\hat{v}_o(s)}{\hat{d}(s)}\right|_{\hat{v}_g=0,\hat{i}_o=0} = \frac{\left(\frac{V_o}{D'} - r'_L I_L\right) - sL' I_L}{1 + s r'_L C + s^2 L' C}$$

$r'_L = \frac{r_L}{D'^2}$, $L' = \frac{L}{D'^2}$.

Interpretation with exponential response, step response and frequency domain

State-space averaging: time variant model

State-space averaging is a generalization of the averaged small-signal modeling. Let us consider a converter that evolves between two structures S_0 and S_1. The structure depends on the switches positions. The state-space equations are:

$$\begin{aligned}\frac{d\boldsymbol{x}}{dt} =& \boldsymbol{A}_c\boldsymbol{x}(t) + \boldsymbol{B}_c\boldsymbol{v}(t),\\ \boldsymbol{y}(t) =& \boldsymbol{C}_c\boldsymbol{x}(t) + \boldsymbol{E}_c\boldsymbol{v}(t).\end{aligned}$$

$\boldsymbol{x}$, $\boldsymbol{v}$ and $\boldsymbol{y}$ represent respectively the state, input and output vectors. $\boldsymbol{A}_c$, $\boldsymbol{B}_c$, $\boldsymbol{C}_c$, $\boldsymbol{E}_c$ are matrices that model the converter for each switch position $c \in \{0, 1\}$. Let us now use define the PWM signal $c(t)$ and its complement $c'(t) = 1 - c(t)$. We can now rewrite the above equations.
Side note: in the LSD course $\boldsymbol{E}$ matrix is named $\boldsymbol{D}$. We keep here the $\boldsymbol{E}$ notation to avoid confusion with the duty-cycle.

Example: see [1] eq 1.39, 1.40, 1.41, 1.42, 1.43, 1.44

State-space averaging: time invariant model

$$
\begin{aligned}
\frac{d\boldsymbol{x}}{dt} =& c(t)[\boldsymbol{A}_1\boldsymbol{x}(t) + \boldsymbol{B}_1\boldsymbol{v}(t)] + c'(t)[\boldsymbol{A}_0\boldsymbol{x}(t) + \boldsymbol{B}_0\boldsymbol{v}(t)],\\
\boldsymbol{y}(t) =& c(t)[\boldsymbol{C}_1\boldsymbol{x}(t) + \boldsymbol{E}_1\boldsymbol{v}(t)] + c'(t)[\boldsymbol{C}_0\boldsymbol{x}(t) + \boldsymbol{E}_0\boldsymbol{v}(t)].
\end{aligned}
$$

We can apply the averaging operator $\langle . \rangle_{T_s}$ on both sides of the equation and with the small ripple approximation, we get the **averaged large-signal state-space equations**:

$$
\begin{aligned}
\frac{d\bar{\boldsymbol{x}}}{dt} =& [d(t)\boldsymbol{A}_1 + d'(t)\boldsymbol{A}_0]\bar{\boldsymbol{x}}(t) + [d(t)\boldsymbol{B}_1 + d'(t)\boldsymbol{B}_0]\bar{\boldsymbol{v}}(t),\\
\bar{\boldsymbol{y}} =& [d(t)\boldsymbol{C}_1 + d'(t)\boldsymbol{C}_0]\bar{\boldsymbol{x}}(t) + [d(t)\boldsymbol{E}_1 + d'(t)\boldsymbol{E}_0]\bar{\boldsymbol{v}}(t).
\end{aligned}
$$

Thanks to the averaging, the time varying nature of the system has been removed but the equations are still non-linear.

State-space averaging: operating point

The operating point can be found by solving the above equations for $\frac{d\bar{\boldsymbol{x}}}{dt} = 0$:

$$\begin{aligned}\mathbf{0} =& [D\boldsymbol{A}_1 + D'\boldsymbol{A}_0]\boldsymbol{X} + [D\boldsymbol{B}_1 + D'\boldsymbol{B}_0]\boldsymbol{V},\\ \boldsymbol{Y} =& [D\boldsymbol{C}_1 + D'\boldsymbol{C}_0]\boldsymbol{X} + [D\boldsymbol{E}_1 + D'\boldsymbol{E}_0]\boldsymbol{V}.\end{aligned}$$

With the following definition,

$$\begin{aligned}\boldsymbol{A} \triangleq& D\boldsymbol{A}_1 + D'\boldsymbol{A}_0,\ \ \boldsymbol{B} \triangleq D\boldsymbol{B}_1 + D'\boldsymbol{B}_0,\\ \boldsymbol{C} \triangleq& D\boldsymbol{C}_1 + D'\boldsymbol{C}_0,\ \ \boldsymbol{E} \triangleq D\boldsymbol{E}_1 + D'\boldsymbol{E}_0,\end{aligned}$$

we get:

$$\boldsymbol{X} = -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{V},\ \ \boldsymbol{Y} = [-\boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B} + \boldsymbol{E}]\boldsymbol{V}.$$

The above solution is equivalent to apply the inductors volt-second balance and the capacitors charge balance under the small-ripple approximation. Example: see [1] eq 1.45, 1.46, 1.47

State-space averaging: small signal model

The state equation can be linearized by defining small signals around the operating point:

$$\hat{\bar{\boldsymbol{x}}}(t) \triangleq \bar{\boldsymbol{x}}(t) - \boldsymbol{X},\ \hat{d} \triangleq d(t) - D,\ \hat{\bar{\boldsymbol{v}}}(t) \triangleq \bar{\boldsymbol{v}}(t) - \boldsymbol{V}.$$

Introducing the above definitions in the averaged large-signal state-space equations, we get the **small-signal equations**:

$$\begin{aligned}\frac{d\hat{\bar{\boldsymbol{x}}}}{dt} =& \boldsymbol{A}\hat{\bar{\boldsymbol{x}}}(t) + \boldsymbol{F}\hat{d}(t) + \boldsymbol{B}\hat{\bar{\boldsymbol{v}}}(t),\\ \hat{\bar{\boldsymbol{y}}}(t) =& \boldsymbol{C}\hat{\bar{\boldsymbol{x}}}(t) + \boldsymbol{G}\hat{d}(t) + \boldsymbol{E}\hat{\bar{\boldsymbol{v}}}(t),\end{aligned}$$

where,

$$\begin{aligned}\boldsymbol{F} \triangleq& (\boldsymbol{A}_1\boldsymbol{X} + \boldsymbol{B}_1\boldsymbol{V}) - (\boldsymbol{A}_0\boldsymbol{X} + \boldsymbol{B}_0\boldsymbol{V}),\\ \boldsymbol{G} \triangleq& (\boldsymbol{C}_1\boldsymbol{X} + \boldsymbol{E}_1\boldsymbol{V}) - (\boldsymbol{C}_0\boldsymbol{X} + \boldsymbol{E}_0\boldsymbol{V}).\end{aligned}$$

State-space averaging: solving the small signal model

We can solve the small-signal equations in the Laplace domain:

$$\begin{aligned} s\hat{\boldsymbol{x}}(s) =& \boldsymbol{A}\hat{\boldsymbol{x}}(s) + \boldsymbol{F}\hat{d}(s) + \boldsymbol{B}\hat{\boldsymbol{v}}(s), \\ \hat{\boldsymbol{y}}(s) =& \boldsymbol{C}\hat{\boldsymbol{x}}(s) + \boldsymbol{G}\hat{d}(s) + \boldsymbol{E}\hat{\boldsymbol{v}}(s), \\ \Rightarrow \hat{\boldsymbol{y}}(s) =& (\boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{F} + \boldsymbol{G})\hat{d}(s) + (\boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{E})\hat{\boldsymbol{v}}(s) \end{aligned}$$

The **control transfer matrix** is:

$$\boldsymbol{W}(s) \triangleq \left.\frac{\hat{\boldsymbol{y}}(s)}{\hat{d}(s)}\right|_{\hat{\boldsymbol{v}}(s)=0} = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{F} + \boldsymbol{G}.$$

The **disturbance transfer matrix** is:

$$\boldsymbol{W}_D(s) \triangleq \left.\frac{\hat{\boldsymbol{y}}(s)}{\hat{\boldsymbol{v}}(s)}\right|_{\hat{d}(s)=0} = \boldsymbol{C}(s\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{E}.$$

Example: see [1] eq 1.48, 1.49, the boost transfer function has a RHZ (Right Half-Plane Zero).

State-space averaging: solving the small signal model

- Link with the LSD course: each term of $\boldsymbol{W}(s)$ and $\boldsymbol{W}_D(s)$ represents the transfer function that models the effect of external inputs (duty-cycle, input voltage....) on outputs. Because there are multiple outputs and multiple inputs the system is called MIMO.

The pulse width modulator

In order to transform the duty-cycle (continuous variable that has a value between 0 and 1) into binary (”ON/OFF”) signals that control power switches, we need a building block called ”modulator”.

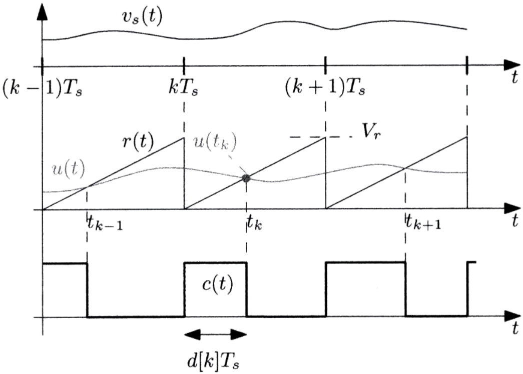
There are two main families of PWM modulators:

NSPWM: naturally sampled pulse width modulator. They process a continuous time modulating signal $u(t)$. They are typically used in analog controllers.

USPWM: uniformly sampled pulse width modulator. They process a sampled signal $u[k]$ and generate a PWM signal updated every switching period. They are typically used in digital controllers.

Naturally sampled pulse width modulator

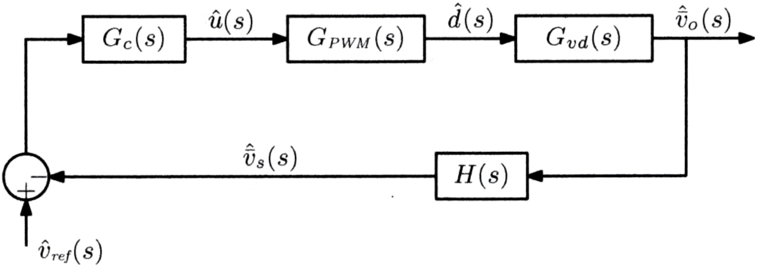
Excerpt of [1]:



$$d[k] = \frac{u(t_k)}{V_r} \Rightarrow G_{PWM}(s) \triangleq \frac{\hat{d}}{\hat{u}} = \frac{1}{V_r} \qquad (1)$$

Closed loop system

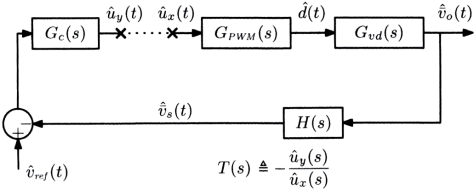
Excerpt of [1]:



The above figure shows a block diagram of a closed loop system. $G_{PWM}(s)$ is the transfer function of the PWM modulator. $G_{vd}(s)$ models the converter behavior. $G_c(s)$ is the compensator function to be designed. $H(s)$ is the output voltage (current) sensor transfer function.

Definition of the loop gain

Excerpt of [1]:



$$T(s) \triangleq -\left.\frac{\hat{u}_y(s)}{\hat{u}_x(s)}\right|_{\hat{v}_{ref}=0} = G_c(s)G_{PWM}(s)G_{vd}(s)H(s)$$

$$\boxed{T_u(s) \triangleq G_{PWM}(s)G_{vd}(s)H(s)} \tag{2}$$

For the buck converter, we obtain:

$$T_u(s) = \frac{1}{V_r}G_{vd0}\frac{1+\frac{s}{\omega_{ESR}}}{1+\frac{s}{Q\omega_0}+\frac{s^2}{\omega_0^2}}H(s).$$

Link with LSD course

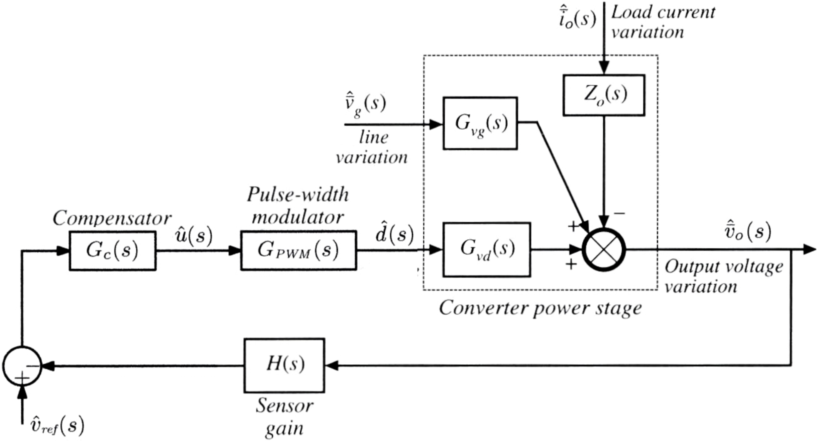
- In the LSD course, the system to be controlled is called the plant and has a transfer function $P(s)$. In a power converter, the plant consists of the PWM modulator, the power electronics circuit and the measurement circuit:

$$P(s) = G_{PWM}(s)G_{vd}(s)H(s) \tag{3}$$

- In the LSD course, the controller has a transfer function $C(s)$ that is named here $G_c(s)$.
- In the LSC course, the loop gain is called $L(s)$, here it is called $T(s)$.

Loop including external perturbation

Excerpt of [1]:



Closed loop gain

- The uncompensated loop **phase margin** (at the crossover frequency) gives a stability criteria and allows to design the compensator $G_c(s)$.
- The reference setpoint to the output transfer function is given by:

$$G_{vvref,cl}(s) \triangleq \left.\frac{\hat{\hat{v}}_o(s)}{\hat{\hat{v}}_{ref}}\right|_{\hat{\hat{v}}_g(s)=0,\hat{\hat{i}}_o=0} \qquad = \frac{1}{H(s)}\frac{T(s)}{1+T(s)},$$

Closed loop gain

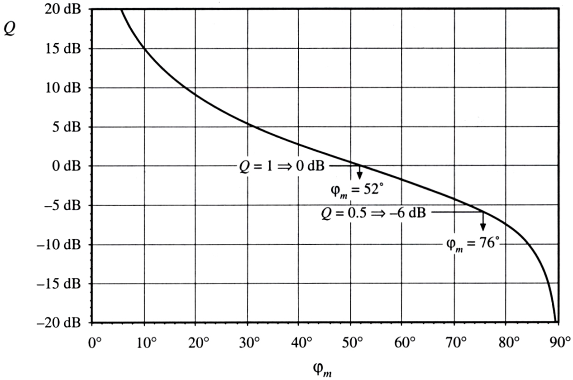
- The sensitivity characteristics to external perturations is reduced by increasing the loop gain. Refering to the figure on the previous slide, the **closed loop** characteristics can be derived from the **open loop** characteristics:

$$
\begin{aligned}
G_{vg,cl}(s) &\triangleq \left.\frac{\hat{\hat{v}}_o(s)}{\hat{\hat{v}}_g(s)}\right|_{\hat{\hat{v}}_{ref}=0,\hat{\hat{i}}_o=0} &= \frac{G_{vg}(s)}{1+T(s)},\\
Z_{o,cl}(s) &\triangleq -\left.\frac{\hat{\hat{v}}_o(s)}{\hat{\hat{i}}_o(s)}\right|_{\hat{\hat{v}}_{ref}=0,\hat{\hat{v}}_g=0} &= \frac{Z_o(s)}{1+T(s)}.
\end{aligned}
$$

From the above relations, the goal is to get $T(s)$ as large as possible on a large bandwidth while maintaining a good phase margin.

Relation between phase margin and stability

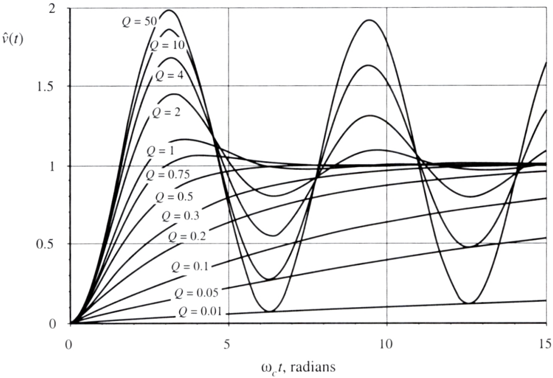
Excerpt of [2]:



For open loop characteristics with a loop gain that falls by -20 dB/decade when $T(s)$ amplitude approaches 1, the phase margin directly affects the quality factor of the closed loop (second order) system response.

Relation between phase margin and stability

Excerpt of [2]:



The step response of the closed loop (second order) system is directy related to the quality factor. The choice of a phase margin of 52° is now explained.

Analog control loop design procedure

1. Determine $G_{PWM}(s)$ using equation 1.
2. Determine $H(s)$ based on the specifications/design of your sensor.
3. Detemine the transfer function of your converter with the presented modeling techniques.
4. Trace the Bode Plots (MATALB) for $T_u(s)$ (equation 2) for different operating points (input voltage, load).
5. Choose the cross-over frequency f_c typically 1/10 of the switching frequency.
6. Choose the target phase margin ϕ_m typically $> 52^\circ$.
7. Choose your compensator:
   - if DC error has to be canceled, use a PI or PID,
   - if phase margin has to be increased, use a PD or PID,
   - if phase margin is already 90°, use a P or PI.
8. For the integrator term, choose a corner frequency that is 1/10 of the choosen cross-over frequency f_c.

Chapter 2: The Digital Control Loop

- Analog vs. digital control
    - Example: digital voltage-mode control
    - A/D conversion
- The digital compensator
    - The digital PID controller (additive form)
    - Bilinear mapping
    - PID transformation with the bilinear mapping
    - PID controller in multiplicative form
- The digital pulse width modulator
- Loop delays
- Digital control loop design procedure
- The integral windup problem
- Unit conversion

Analog vs. digital control

Digital control of switched mode converters introduces two differences in comparison to analog control:

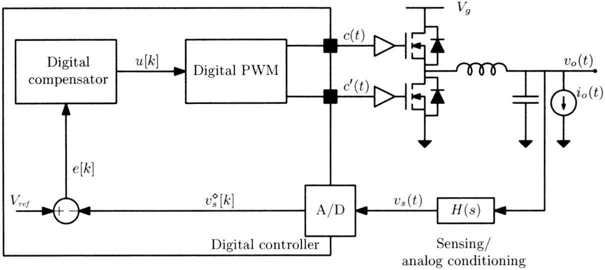
Time quantization: the controller samples values of analog variables, processes them to evaluate the modulation (duty-cycle) and apply it for one sampling period.

Amplitude quantization: analog variables are sampled with finite resolution analog-to-digital converters, they are therefore quantized.

There are different approaches to model switching converters. The approach presented here is based on the averaged model.

Example: digital voltage-mode control

Excerpt of [1]:



The sampled signal is defined by:

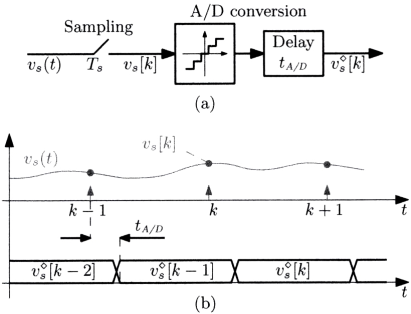
$$v_s[k] \triangleq v_s(t_k).$$

The most common choice for the **sampling period** is :

$$\boxed{T = T_s}, \quad \text{where } T_s \text{ is the switching period.}$$

A/D conversion

Excerpt of [1]:



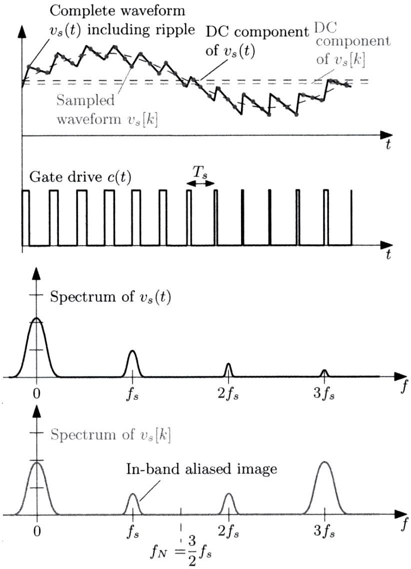
- Sampling process
- Amplitude quantization
- Conversion delay $t_{A/D}$

The sampling process moves the modeling problem from the analog to the digital world. The amplitude quantization makes the problem non linear. The delay modifies the dynamics.

Example: show captured data.

Sampling rate different from f_s

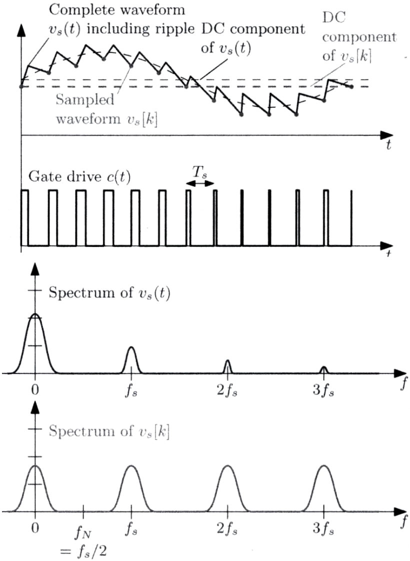
Excerpt of [1]:



- Alias of the high frequency content of the analog signal is present in the sampled signal (for example at f_s).
- Large digital filtering efforts are therefore required.

Sampling rate equal to f_s

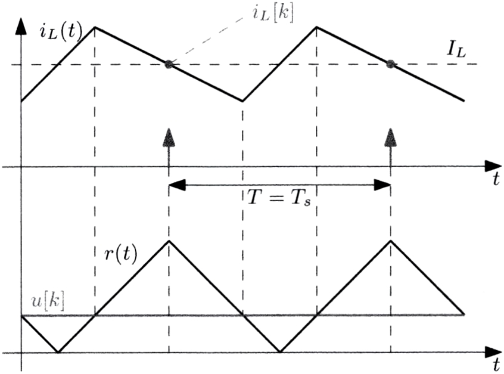
Excerpt of [1]:



- Alias of the high frequency content of the analog signal is present in the sampled signal but only at DC.
- No filtering efforts is needed, only DC compensation is required.

Sampling strategy to avoid DC alias

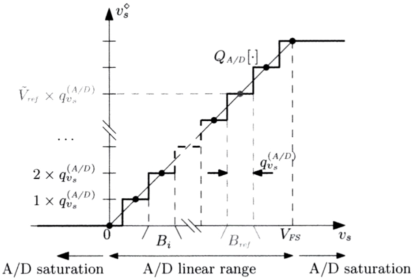
Excerpt of [1]:



For triangular waveforms, the most common solution is to sample the analog signal in the middle of the ramp. This suppresses DC aliasing effect.

Amplitude quantization

Excerpt of [1]:



The A/D converter linear range is divided into $2^{n_{A/D}}$ bins. Each bin is $q_{v_s}^{(A/D)}$ volts wide:

$$q_{v_s}^{(A/D)} = \frac{V_{FS}}{2^{n_{A/D}}} \qquad (4)$$

where V_{FS} is the full scale voltage.

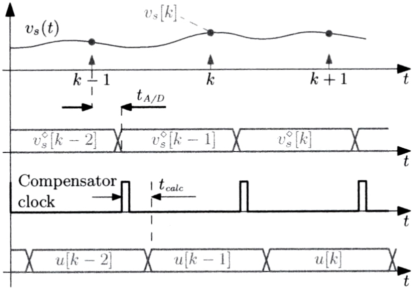
The figure shows the quantization characteristic $Q_{A/D}[.]$:

$$v_s^{\diamond}[k] \triangleq Q_{A/D}[v_s[k]] = q_{v_s}^{(A/D)} \tilde{v}_s[k]$$

where $v_s^{\diamond}[k]$ is the quantized signal, $v_s[k]$ is the analog signal and $\tilde{v}_s[k]$ is the binary coded signal.

The controller

Excerpt of [1]:



The analog signal $v_s(t)$ is sampled and quantized in $v_s^\diamond[k]$ after the $t_{A/D}$ conversion delay . The compensator uses this sampled signal to generate the new PWM command $u[k]$ after the calculation delay t_{calc}.

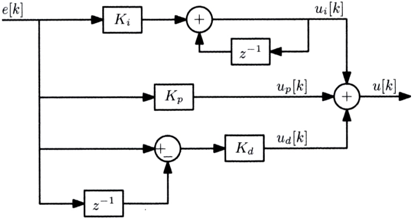
A **linear** and **time-invariant** compensation law is described by a difference equation:

$$\begin{aligned} u[k] = & -a_1 u[k-1] - a_2 u[k-2] - \ldots - a_N u[k-N] \\ & + b_0 e[k] + b_1 e[k-1] + b_2 e[k-2] + \ldots + b_M e[k-M] \end{aligned}$$

The PID compensation law is a praticular case of the above equation and will be presented.

The digital PID controller (additive form)

Excerpt of [1]:



The PID law is given by:

$$
\begin{aligned}
u_p[k] =& K_p e[k], \\
u_i[k] =& u_i[k-1] + K_i e[k], \\
u_d[k] =& K_d(e[k] - e[k-1]), \\
u[k] =& u_p[k] + u_i[k] + u_d[k].
\end{aligned}
$$

The z-transform of the above laws gives the transfer function of the PID compensator:

$$G_{PID}(z) \triangleq \frac{U(z)}{E(z)} = K_p + \frac{K_i}{1 - z^{-1}} + K_d(1 - z^{-1})$$

How can we determine the digital coefficients K_p, K_i and K_d?

Bilinear mapping

The coefficients can be determined by using the classical compensation techniques in the s-domain. For that purpose the $G_{PID}(z)$ function can be transformed using:

$$z = e^{sT} \tag{5}$$

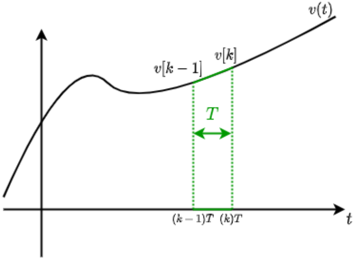
This transformation is not adequate because it transforms $G_{PID}(z)$ in a transcendental function of s. A good approximation is:

$$z(s) = e^{sT} = \frac{e^{+s\frac{T}{2}}}{e^{-s\frac{T}{2}}} = \frac{1 + s\frac{T}{2} + \cdots}{1 - s\frac{T}{2} + \cdots} \approx \frac{1 + s\frac{T}{2}}{1 - s\frac{T}{2}} \tag{6}$$

It is called the bilinear transformation:

$$\boxed{z(s) \approx \frac{1 + s\frac{T}{2}}{1 - s\frac{T}{2}} \;\Leftrightarrow\; s(z) \approx \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}}} \tag{7}$$

Bilinear mapping interpretation: trapezoidal approximation



In time domain:

$$\int v(\tau)d\tau = p(t)$$

In s-domain:

$$\frac{1}{s}V(s) = P(s) \Rightarrow V(s) = s\ P(s)$$

We can write in discrete time:

$$p[k] - p[k-1] = \textit{trapeze surface} = \frac{v[k-1] + v[k]}{2}T$$

In z-domain:

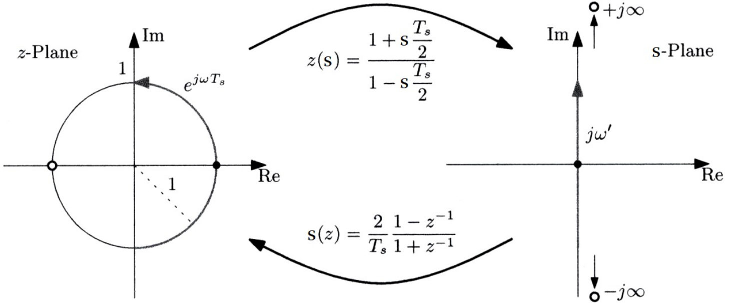
$$P(z) - z^{-1}P(z) = \frac{z^{-1}V(z) + V(z)}{2}T \Rightarrow V(z) = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}P(z)$$

Bilinear mapping

The bilinear mapping has the following properties:

- It is a rational transformation.
- Stability limits are conserved: the unit circle ($|z| = 1$) in the z-domain is mapped on the y axis in the s-domain.
- Some frequency wrapping is introduced (due to the approximation) but it yields less than 10% error for frequencies below $\frac{1}{6}\frac{1}{T}$.

Excerpt of [1]:


z-Plane
Im
1
$e^{j\omega T_s}$
Re
1
$z(s) = \frac{1 + s\frac{T_s}{2}}{1 - s\frac{T_s}{2}}$
$s(z) = \frac{2}{T_s}\frac{1 - z^{-1}}{1 + z^{-1}}$
$+j\infty$
Im
s-Plane
$j\omega'$
Re
$-j\infty$

PID transformation with the bilinear mapping

Application of the bilinear transformation allows us to work in the s-domain and to use the classical analog design tools.

$$G_{PID}(z) \triangleq \frac{U(z)}{E(z)} = K_p + \frac{K_i}{1 - z^{-1}} + K_d(1 - z^{-1})$$

is transformed to

$$G'_{PID}(s) \triangleq \frac{U(s)}{E(s)} = K_p + \frac{K_i}{T}\frac{1 + \frac{s}{\omega_p}}{s} + K_d T \frac{s}{1 + \frac{s}{\omega_p}},$$

where $\omega_p \triangleq \frac{2}{T}$.
It should be noted that ω_p appears when converting $G_{PID}(z)$ in the s-domain and there is no freedom on the value of ω_p.

PID transformation with the bilinear mapping

- On the previous slide, K_i is the digital coefficient of the integrator. In the s-domain, it is **divided** by T. This can be explained physically as follows: if T is for example increased, the digital accumulation ($u_i[k] = u_i[k-1] + K_i e[k]$) will be performed less often due to the larger sampling period T. This is equivalent to a slower integral in the s-domain that is represented by a lower analog integrator gain.
- In a similar way, K_d is the digital coefficient for the derivative part. When transformed in the s-domain, it is **multiplied** by T. This can be explained physically as follows: if T is for example increased, the digital derivative term ($u_d[k] = K_d(e[k] - e[k-1])$) will performed on a larger sampling period T. Therefore, the estimation of the error variation will be taken on sample $e[k]$ and $e[k-1]$ that are more spaced in time. This will amplify the value of $e[k] - e[k-1]$ which is equivalent to a multiplication by T in the s-domain.
- The multiplicative form of the PID compensator is easier to use and equivalence relation exists (see next slides).

PID controller in multiplicative form

$$G'_{PID}(s) = K_p + \frac{K_i}{T}\frac{1+\frac{s}{\omega_p}}{s} + K_d T \frac{s}{1+\frac{s}{\omega_p}} \tag{8}$$

is equivalent to

$$G'_{PID}(s) = G'_{PI\infty}(1+\frac{\omega_{PI}}{s})G'_{PD0}\frac{1+\frac{s}{\omega_{PD}}}{1+\frac{s}{\omega_p}}$$

with,

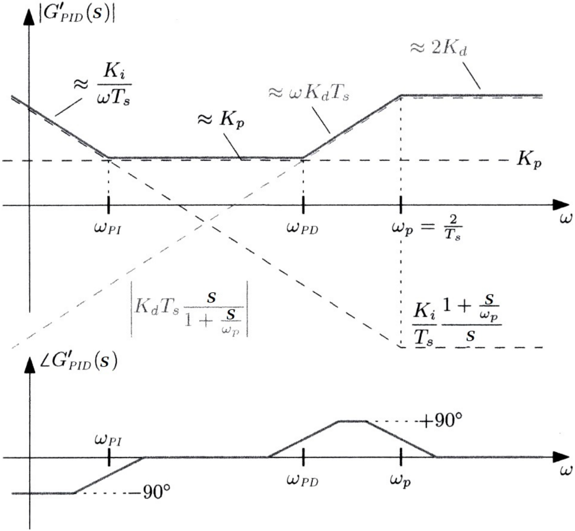
$$K_p = G'_{PI\infty}G'_{PD0}(1+\frac{\omega_{PI}}{\omega_{PD}}-\frac{2\omega_{PI}}{\omega_p}), \tag{9}$$

$$K_i = 2G'_{PI\infty}G'_{PD0}\frac{\omega_{PI}}{\omega_p}, \tag{10}$$

$$K_d = \frac{1}{2}G'_{PI\infty}G'_{PD0}(1-\frac{\omega_{PI}}{\omega_p})(\frac{\omega_p}{\omega_{PD}}-1). \tag{11}$$

PID bode plot

Excerpt of [1]:



PI (lag) compensator in multiplicative form

$$G'_{PI}(s) = K_p + \frac{K_i}{T}\frac{1+\frac{s}{\omega_p}}{s} \tag{12}$$

is equivalent to

$$G'_{PI}(s) = G'_{PI\infty}(1+\frac{\omega_{PI}}{s})$$

with,

$$K_p = G'_{PI\infty}(1-\frac{\omega_{PI}}{\omega_p}), \tag{13}$$

$$K_i = 2G'_{PI\infty}\frac{\omega_{PI}}{\omega_p}. \tag{14}$$

PD (lead) compensator in multiplicative form

$$G'_{PD}(s) = K_p + K_d T \frac{s}{1 + \frac{s}{\omega_p}} \tag{15}$$

is equivalent to

$$G'_{PD}(s) = G'_{PD0} \frac{1 + \frac{s}{\omega_{PD}}}{1 + \frac{s}{\omega_p}}$$

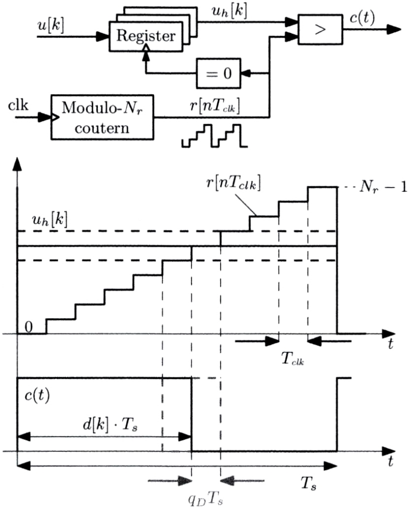
with,

$$K_p = G'_{PD0}, \tag{16}$$

$$K_d = \frac{1}{2} G'_{PD0} (\frac{\omega_p}{\omega_{PD}} - 1). \tag{17}$$

The digital pulse width modulator

Excerpt of [1]:



Digital modulator are based on a digital counter. The higher the counter clock (T_{clk}), the higher the resolution.

The behavior of the digital PWM is given by:

$$T_s = N_r T_{clk}, \tag{18}$$

$$d[k] = \frac{u[k]}{N_r}, \tag{19}$$

$$q_D = \frac{T_{clk}}{T_s} = \frac{1}{N_r}. \tag{20}$$

Loop delays

Different delays exist between the sampling of the analog signal up to the generation of the PWM:

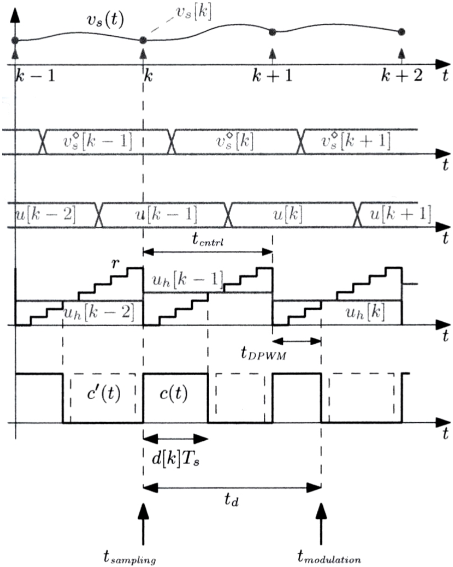
- Control delay (t_{cntrl}):
  - FPGA based controller: all calculations are performed in //, the processing delay is negligible and the only delay is the A/D conversion delay occuring between the sampling of the analog signal and the availability of the sampled version $v_s^{\diamond}[k]$.
  - DSP (CPU) based controller: several instructions are needed and there is an extra delay to be taken into account.
- Modulation delay (t_{DPWM}): the PWM modulator has an intrinsic delay due to the PWM process itself.

The total delay induced by the control process is modeled:

$$t_d = t_{cntrl} + t_{DPWM} \;\Rightarrow\; e^{-st_d}.$$

Loop delays

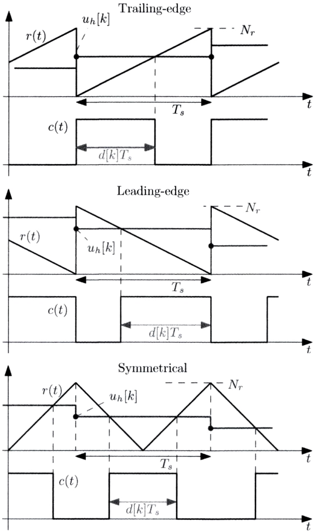
Excerpt of [1]:



A typical software based controller is shown.
The A/D sampling and the controller calculations are performed during one switching period ($= t_{cntrl}$).
The PWM delay (t_{DPWM}) occurs after the control delay.

DPWM delay

Excerpt of [1]:



$$G_{PWM,TE}(j\omega) = \frac{e^{-j\omega DT_s}}{N_r}$$

$$G_{PWM,LE}(j\omega) = \frac{e^{-j\omega(1-D)T_s}}{N_r}$$

$$G_{PWM,Sym}(j\omega) =$$

$$= \frac{\cos\left(\omega DT_s/2\right)}{N_r} e^{-j\omega\frac{T_s}{2}}$$

$$\approx \frac{e^{-j\omega\frac{T_s}{2}}}{N_r}$$

Total loop delay

The total loop delay is defined by:

$$\boxed{t_d \triangleq t_{cntrl} + t_{DPWM}.}$$

This total delay is modeled in the s-domain with e^{-st_d}. Before estimating compensation factors, the loop gain is corrected to take this delay into account:

$$\boxed{T_u^{\dagger}(s) \triangleq T_u(s)e^{-st_d}.}$$

Digital control loop design procedure

1. Model the loop as in the analog control loop design but with the delay corrected loop gain $T_u^{\dagger}(s)$.
2. Design the compensation as in the analog control loop design using the chosen compensator (equation 8, 12 or 15).
3. Once the controller coefficient are determined, transform them to their digital version (equation 9 to 11, 13 to 14 or 16 to 17).
4. Implement the control law in the digital processor.

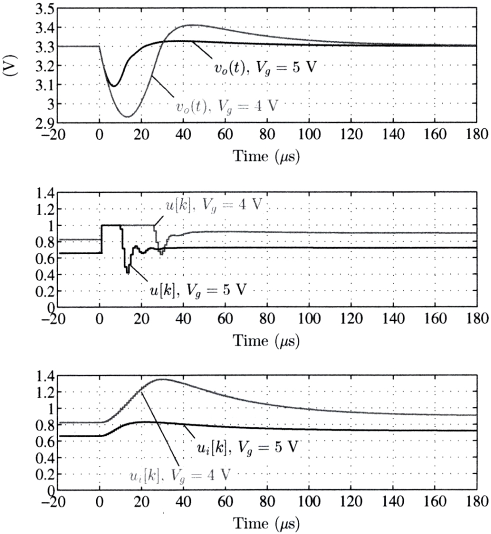
It should be noted that the above design procedure is valid if the sampled signal is a good representation of the averaged signal. This assumption is the *small-aliasing approximation* expressed mathematically by:

$$v_s[k] \approx \bar{v}_s(t_k).$$

If it is not the case, discrete-time modeling techniques have to be used.

The integral windup problem (example)

Excerpt of [1]:



For $V_g = 4V$, at $0\mu s$, the current rises to 10 A. The controller reacts quickly to the induced $v_o(t)$ change and $u[k]$ quickly reaches its saturation point. However, $v_o(t)$ is still under the set point and the integrator continues to integrate. When $v_o(t)$ starts to rise again, the integrator is well above 1 and forces $u[k]$ to stay at 1 and creates an unexpected lag in the reaction of the crontroller finally creating a $v_o(t)$ overshoot.

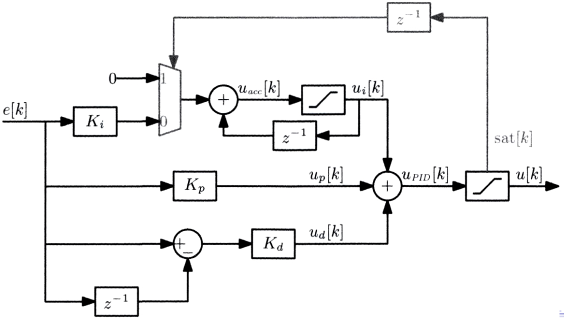
Actuator saturation and integral anti-windup

The solution to the windup problem is to:

- Saturate all variables to avoid numeric issues. Especially saturate $u[k]$ between the min/max duty-cycle.
- Stop integration when $u[k]$ reaches its saturation limits:

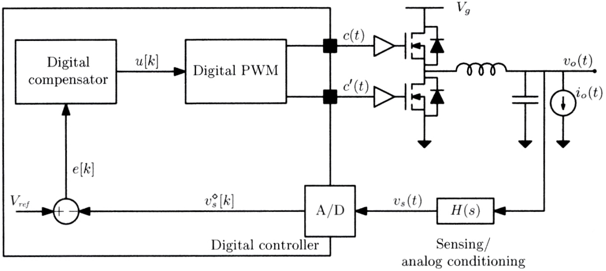
$$\text{sat}[k] = \begin{cases} 0 & \text{if } 0 \leq u_{PID}[k] \leq 1, \\ 1 & \text{otherwise.} \end{cases}$$

Excerpt of [1]:



Unit conversion

Excerpt of [1]:



- K_p, K_i and K_d on slide "The PID compensator (additive form)" have physical units: $\frac{1}{V}$.
- In the processor, $v_s^\diamond[k]$ and $u[k]$ are represented by integer numbers related to the respective resolutions $2^{n_{A/D}}$ (equation 4) and N_r (equation 20).

Unit conversion: from physical to digital representation

	Design (physical unit)	Digital representation
$u[k]$	$0 \rightarrow 1$	$0 \rightarrow N_r$
$v_s[k]$	$0 \rightarrow V_{FS}$	$0 \rightarrow 2^{n_{A/D}}$
$K_p,\ K_i,\ K_d$	in $\frac{1}{V}$	multiply by $\frac{V_{FS}}{2^{n_{A/D}}} N_r$

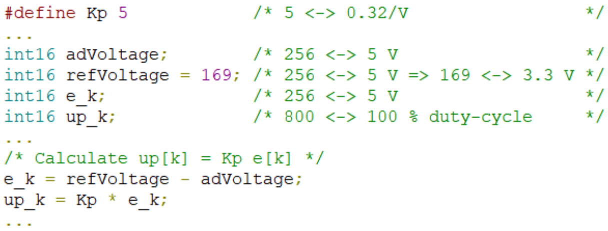
Table: Conversion from physical to digital representation.

- The above table is also valid for current measurements because the current is transformed in a voltage by the measurement device. This is modeled by $H(s)$.
- K_i is often lower than K_p yielding to quantification errors. $u_i[k] = u_i[k-1] + K_i e[k]$ can be performed in two steps:

$$accumulator[k] = accumulator[k-1] + e[k],$$
$$u_i[k] = K_i accumulator[k]$$

Unit conversion: example

- The PWM generator runs at $16MHz$ and generate a $20kHz$ signal $\Rightarrow N_r = \frac{16MHz}{20kHz} = 800$.
- ADC of 8 bits resolution over $5V \Rightarrow V_{FS} = 5V$ and $2^{n_{A/D}} = 256$.
- $Kp = 0.32/V \Rightarrow$ digital value of K_p is $K_p \frac{V_{FS}}{2^{n_{A/D}}} N_r = 5$.



```
#define Kp 5            /* 5 <-> 0.32/V                 */
...
int16 adVoltage;        /* 256 <-> 5 V                  */
int16 refVoltage = 169; /* 256 <-> 5 V => 169 <-> 3.3 V */
int16 e_k;              /* 256 <-> 5 V                  */
int16 up_k;             /* 800 <-> 100 % duty-cycle     */
...
/* Calculate up[k] = Kp e[k] */
e_k = refVoltage - adVoltage;
up_k = Kp * e_k;
...
```

- Clearly document the variables regarding their physical meaning.
- Multiplication by K_p, K_i, K_d can often be replaced by shift.

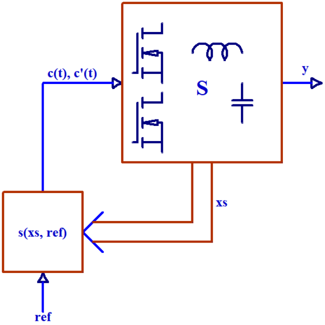
Hands-on: The complete design process

- Controller design (scripts) with some approximations:
  - averaging,
  - linearization.
- Controller verification (Spice based or HIL simulators) with a more accurate model:
  - time variance,
  - non linearities (including quantization).

Current Mode Control

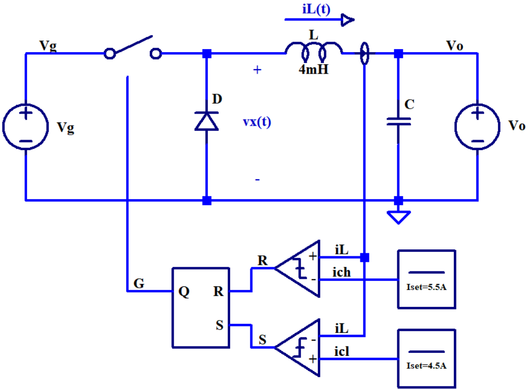
- Current mode control: general principle
- Hysteretic control
  - Principle
  - Operating frequency
- Current programmed control
  - Principle
  - Stability analysis
  - Compensation
- External voltage loop

Current mode control: general principle



- The plan to be controlled is a variable structure circuit.
- Current mode control takes advantage of the variable structure nature of the circuit.
- The $s(x_s, ref)$ function directly changes the circuit structure by acting on the switches state.
- Current mode control is a special case of a sliding mode controller (see [3]).
- The PWM modulator is replaced here by a direct action on the circuit structure based on a function of the state of the system and the reference.

Hysteretic control: principle

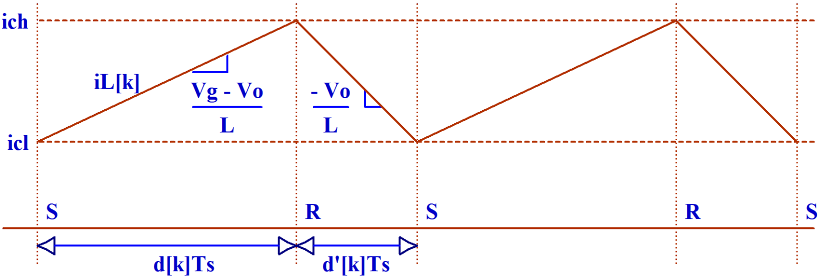


- $i_L(s)$ = state of the system
- s function = comparison with i_{ch}, i_{cl}

- The hysteresis $i_{ch} - i_{cl}$ is introduced to avoid infinite switching frequency.

Example: see 1_HystereticControl.asc

Hysteretic control: operating frequency



$$T_s = d[k]T_s + d'[k]T_s = (i_{ch} - i_{cl})\frac{L}{V_g - V_o} + (i_{ch} - i_{cl})\frac{L}{V_o}$$

$$= (i_{ch} - i_{cl})L\left(\frac{1}{V_g - V_o} + \frac{1}{V_o}\right) = (i_{ch} - i_{cl})L\frac{V_g}{(V_g - V_o)V_o}$$

$$= (i_{ch} - i_{cl})\frac{L}{V_g}\frac{1}{(\frac{V_g}{V_g} - \frac{V_o}{V_g})\frac{V_o}{V_g}}$$

$$= (i_{ch} - i_{cl})\frac{L}{V_g}\frac{1}{(1 - D)D}\ ,\ D \triangleq \frac{V_o}{V_g}$$

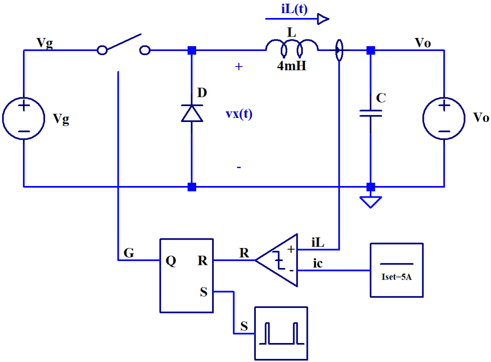
Hysteretic control: discussion

- The inductor average current $i_{cl} + \frac{i_{ch}-i_{cl}}{2} = \frac{i_{ch}+i_{cl}}{2}$
  $\Rightarrow$ The average current in the inductor is perfectly controlled.
- Bipolar operation is natural (current can be > 0 or < 0).
- The switching period is function of V_g and D
  $\Rightarrow$ The switching frequency varies over a large range.

The variation of the switching frequency can be dangerous because the switching frequency can be equal to the resonance frequency of the input or output filter for some operating point. Dangerous resonance can occur.

Therefore, fixed operating frequency is often preferred.

Current programmed control: principle

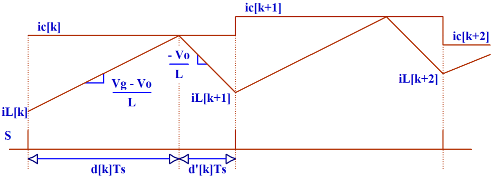


Fixed switching frequency is achieved:

- The switch gate is controlled by a set/reset flip-flop.
- A pulse generator periodically activates the gate at the switching frequency (set).

- The inductor current $i_L(t)$ is permanently compared to the setpoint $i_c(t)$.
- Once inductor current $i_L(t)$ is higher than the setpoint $i_c(t)$, the gate is deactivated (reset).

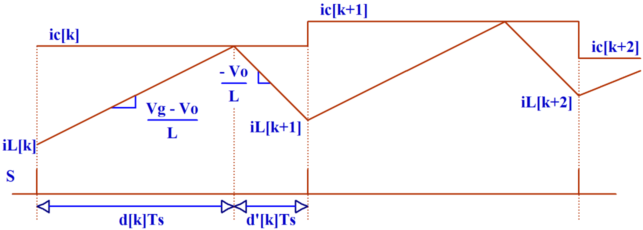
Current programmed control: stability analysis procedure



Instead of using the average model to analyze the control loop, it is possible to directly work in the z-domain. The procedure is as follows:

1. select all state variables ($i_L(t)$),
2. select the sampling instant and name it using "k" index ($i_L[k]$),
3. express the state evolution (matrix form) from "k" to "k+1" ($i_L[k+1]$ as a function of $i_L[k]$, $i_c[k]$),
4. apply linearization ($i_c[k] = I_c + \hat{i}_c[k], i_L[k] = I_L + \hat{i}_L[k]$),
5. linearization gives steady-state and small-signal equations,
6. apply z-transform to small-signal equation by "replacing" $k+1$ by z, $k+2$ by z^2...
7. extract the poles of the z-domain transfer function to access stability,
8. trace the step function and the bode plot to evaluate the performances.

Current programmed control: time domain equation



$$i_c[k] - i_L[k] = d[k]T_s\frac{V_g - V_o}{L} \Rightarrow d[k] = (i_c[k] - i_L[k])\frac{L}{T_s(V_g - V_o)}, \quad (21)$$

$$i_L[k+1] - i_c[k] = d'[k]T_s\frac{-V_o}{L} \Rightarrow i_L[k+1] = i_c[k] - (1 - d[k])T_s\frac{V_o}{L}. \quad (22)$$

Introducing (21) in (22) yields:

$$\boxed{i_L[k+1] = i_c[k](1 + \frac{V_o}{V_g - V_o}) - i_L[k]\frac{V_o}{V_g - V_o} - T_s\frac{V_o}{L}.} \quad (23)$$

Current programmed control: small signal and steady state

Let us define:

$$i_c[k] = I_c + \hat{i}_c[k], \tag{24}$$

$$i_L[k] = I_L + \hat{i}_L[k]. \tag{25}$$

Equation (23) becomes:

$$\hat{i}_L[k+1] = \hat{i}_c[k](1 + \frac{V_o}{V_g - V_o}) - \hat{i}_L[k]\frac{V_o}{V_g - V_o},$$

$$I_L = I_c(1 + \frac{V_o}{V_g - V_o}) - T_s\frac{V_o}{L} - I_L\frac{V_o}{V_g - V_o}.$$

Introducing $D = \frac{V_o}{V_g}$ in the above equations yields:

$$\hat{i}_L[k+1] = \hat{i}_c[k]\frac{1}{1-D} - \hat{i}_L[k]\frac{D}{1-D}\ , \tag{26}$$

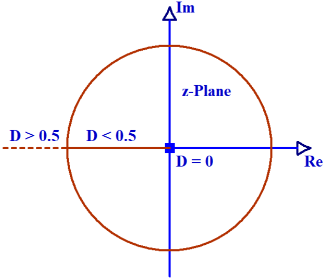
$$I_L = I_c - T_s\frac{V_o}{L}(1-D). \tag{27}$$

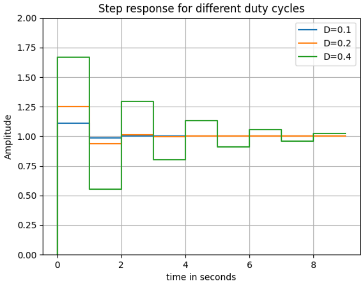
Current programmed control: stability analysis

Taking the z-transform of equation (26):

$$z\hat{I}_L(z) = \hat{I}_c(z)\frac{1}{1-D} - \hat{I}_L(z)\frac{D}{1-D} \ , \tag{28}$$

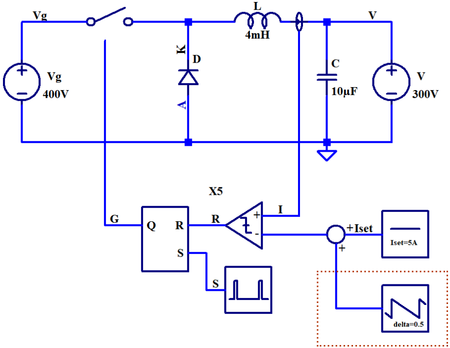
$$\boxed{\hat{I}_L(z) = \hat{I}_c(z)\frac{1}{z(1-D)+D} \ , \ p_z = -\frac{D}{1-D}.} \tag{29}$$





Example: see 2_CurrentProgrammedControl.asc

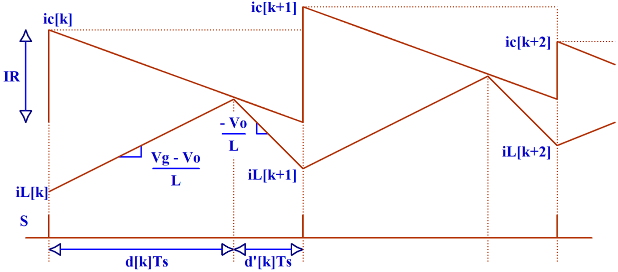
Current programmed control: compensation



Cecil W. Deisch (Bell Laboratories, see [4]) observed that this instability was related to the slope of the current.

- At high duty-cycle, the rising slope is ”too low” and leads to instabilities.
- A small change in current, creates a large change in duty-cycle.
- Compensation is performed by adding an artificial ramp to the current (or by adding the opposite ramp to the reference point, see dotted box on the figure).

Compensation: time domain equation



$$IR \triangleq \delta T_s \frac{V_g}{L}$$

$$i_L[k] + d[k]T_s \frac{V_g - V_o}{L} = i_c[k] - d[k]\delta T_s \frac{V_g}{L} \tag{30}$$

$$\Rightarrow d[k] = (i_c[k] - i_L[k]) \frac{1}{T_s \frac{V_g - V_o}{L} + \delta T_s \frac{V_g}{L}} , \tag{31}$$

$$i_L[k+1] = i_c[k] - d[k]\delta T_s \frac{V_g}{L} - (1 - d[k])T_s \frac{V_o}{L} . \tag{32}$$

Compensation: small signal and steady state

Eliminating $d[k]$ by inserting (31) in (32) yields:

$$\boxed{i_L[k+1] = i_c[k]\left(1+\frac{V_o-\delta V_g}{V_g - V_o+\delta V_g}\right) - i_L[k]\left(\frac{V_o-\delta V_g}{V_g - V_o+\delta V_g}\right) - T_s\frac{Vo}{L}}$$

Using Equation (24) and (25), the previous equation becomes:

$$\begin{aligned}\hat{i}_L[k+1] =& \hat{i}_c[k]\left(1+\frac{V_o-\delta V_g}{V_g - V_o+\delta V_g}\right) - \hat{i}_L[k]\frac{V_o-\delta V_g}{V_g - V_o+\delta V_g} \ ,\\ I_L =& I_c\left(1+\frac{V_o-\delta V_g}{V_g - V_o+\delta V_g}\right) - T_s\frac{V_o}{L} - I_L\frac{V_o-\delta V_g}{V_g - V_o+\delta V_g} .\end{aligned}$$

Introducing $D=\frac{V_o}{V_g}$ in the above equations yields:

$$\hat{i}_L[k+1] = \hat{i}_c[k]\frac{1}{1-(D-\delta)} - \hat{i}_L[k]\frac{D-\delta}{1-(D-\delta)} \ , \tag{33}$$

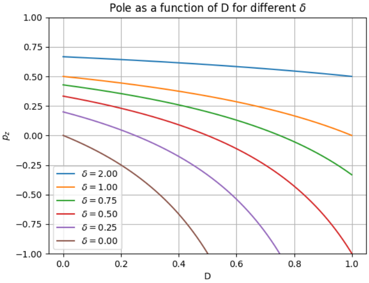
$$I_L = I_c - T_s\frac{V_o}{L}(1-(D-\delta)). \tag{34}$$

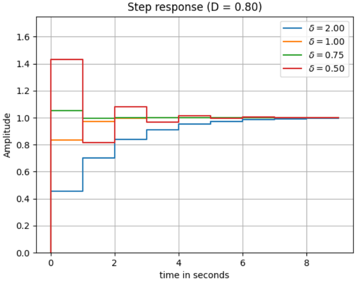
Compensation: stability analysis

Taking the z-transform of equation (33):

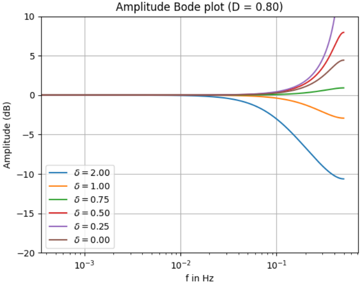
$$z\hat{I}_L(z) = \hat{I}_c(z)\frac{1}{1-(D-\delta)} - \hat{I}_L(z)\frac{D-\delta}{1-(D-\delta)}, \tag{35}$$

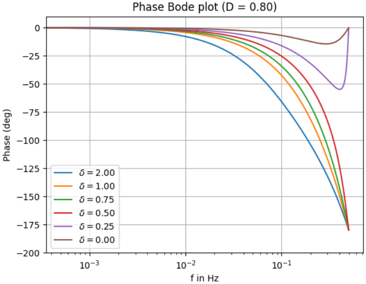
$$\boxed{\hat{I}_L(z) = \hat{I}_c(z)\frac{1}{z(1-(D-\delta))+(D-\delta)}\ ,\ p_z = -\frac{D-\delta}{1-(D-\delta)}.} \tag{36}$$





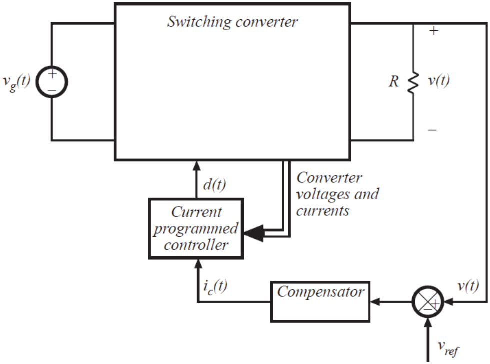
Compensation: discussion





- δ shifts the poles position, $\delta > 0.5$ guarantees stability $\forall D$.
- Poles show the speed of convergence of the remaining error at each switching cycle.
- Therefore, a real negative pole in the z-domain corresponds to sub-harmonic oscillation.
- In practice, when the system is unstable, chaotic behavior appears because large signal operation is non-linear.
- Typically, $\delta \in [0.5, 1.0]$ is enough to achieve stability, too large δ decreases the bandwidth of current mode control.

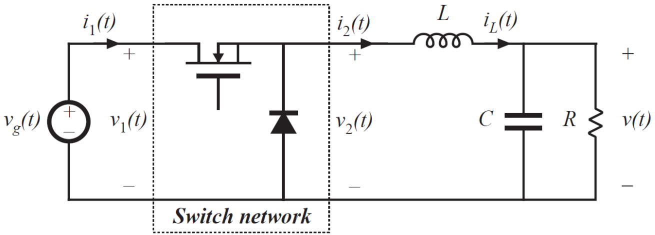
External voltage loop: principle



Excerpt of [2]

- The internal loop is built with the current programmed controller, its setpoint is $i_c(t)$.
- The external control loop computes the voltage error between the set point voltage v_{ref} and the output voltage $v(t)$.
- The voltage error is used by the compensator to drive $i_c(t)$.

External voltage loop: simple approximation



Excerpt of [2]

Over one switching period, the averaged values are related by $d(t)$ in CCM:

$$\begin{aligned}\bar{v_2}(t) =& d(t)\bar{v_1}(t),\\ \bar{i_1}(t) =& d(t)\bar{i_2}(t).\end{aligned}$$

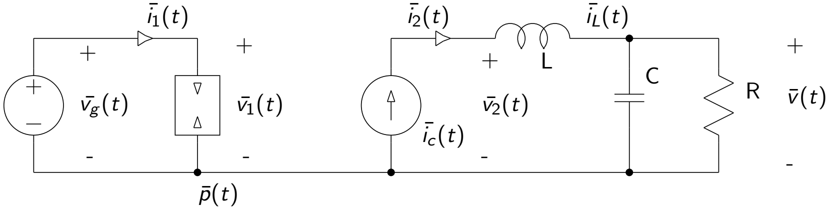
The current control loop is assumed to be perfect (simple approximation):

$$\boxed{\bar{i_2}(t) = \bar{i_c}(t).}$$

$$\Rightarrow \bar{v_1}(t)\bar{i_1}(t) = \bar{v_2}(t)\bar{i_2}(t) = \bar{v_2}(t)\bar{i_c}(t) = \bar{p}(t)$$

The output port is a current source, the input port is a dependent current sink.

Current programmed control: averaged model



The switch is replaced by its averaged model:

- the output port is a controlled current source: $\bar{i}_c(t)$,
- the input port reflects this current with power preservation: $\bar{i}_1(t)\bar{v}_1(t) = \bar{i}_c(t)\bar{v}_2(t) = \bar{p}(t)$.

$\Rightarrow$ linearization has to be applied.

Note: similar reasoning can be applied to other topologies yielding similar models.

Current programmed control: linearized averaged model

$$
\begin{aligned}
\bar{v}_1(t) =& V_1 + \hat{\bar{v}}_1(t),\\
\bar{i}_1(t) =& I_1 + \hat{\bar{i}}_1(t),\\
\bar{v}_2(t) =& V_2 + \hat{\bar{v}}_2(t),\\
\bar{i}_2(t) =& I_2 + \hat{\bar{i}}_2(t),\\
\bar{i}_c(t) =& I_c + \hat{\bar{i}}_c(t).
\end{aligned}
$$

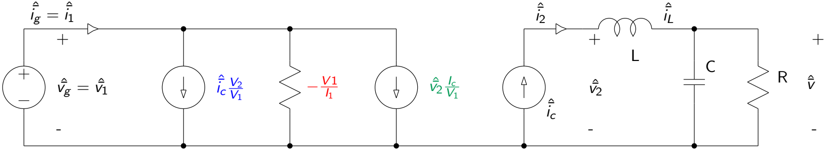
Power conservation can be rewritten:

$$(V_1 + \hat{\bar{v}}_1(t))(I_1 + \hat{\bar{i}}_1(t)) = (V_2 + \hat{\bar{v}}_2(t))(I_c + \hat{\bar{i}}_c(t))$$

The input port current is given by:

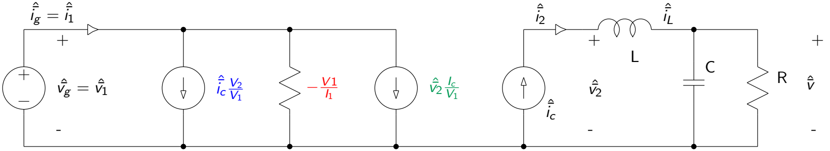
$$\hat{\bar{i}}_1(t) = \hat{\bar{i}}_c(t)\frac{V_2}{V_1} + \hat{\bar{v}}_2(t)\frac{I_c}{V_1} - \hat{\bar{v}}_1(t)\frac{I_1}{V_1}$$

The output port current is: $\hat{\bar{i}}_2(t) = \hat{\bar{i}}_c(t)$



$\hat{\bar{i}}_g = \hat{\bar{i}}_1$
+
$\hat{\bar{v}}_g = \hat{\bar{v}}_1$
-
$\hat{\bar{i}}_c \frac{V_2}{V_1}$
$-\frac{V1}{I_1}$
$\hat{\bar{v}}_2 \frac{I_c}{V_1}$
$\hat{\bar{i}}_c$
$\hat{\bar{i}}_2$
+
$\hat{\bar{v}}_2$
-
L
$\hat{\bar{i}}_L$
C
R
+
$\hat{\bar{v}}$
-

Controlled sources function of input and output voltages

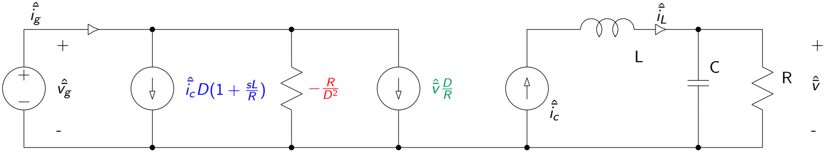


$$\frac{V_2}{V_1} = D, \frac{I_1}{I_2} = D, \frac{V_2}{I_2} = R,$$

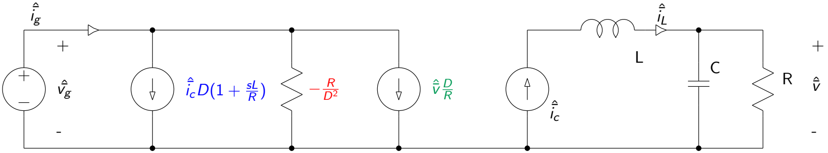
$$-\frac{V_1}{I_1} = -\frac{\frac{V_2}{D}}{I_2 D} = -\frac{R}{D^2},$$

$$\frac{I_c}{V_1} = \frac{I_c}{\frac{V_2}{D}} = \frac{D}{R},$$

$$\hat{v}_2 \frac{I_c}{V_1} = (\hat{v} + sL\hat{i}_c)\frac{D}{R} = \hat{v}\frac{D}{R} + \hat{i}_c D \frac{sL}{R}.$$



Current programmed control: transfer functions



$$G_{vc,cpc}(s) \triangleq \left.\frac{\hat{v}(s)}{\hat{i}_c(s)}\right|_{\hat{v}_g=0} = (R||\frac{1}{sC}) = \frac{R}{1+sRC}$$

→1^{st} order, easy compensator design,

$$G_{vg,cpc}(s) \triangleq \left.\frac{\hat{v}(s)}{\hat{v}_g(s)}\right|_{\hat{i}_c=0} = 0$$

→no sensitivity to input voltage,

$$Z_{g,cpc}(s) \triangleq \left.\frac{\hat{v}_g(s)}{\hat{i}_g(s)}\right|_{\hat{i}_c=0,\hat{v}=0} = -\frac{R}{D^2}$$

→stability concern.

Note: the negative resistance $-\frac{R}{D^2}$ is a **small-signal** negative resistance affecting stability of the source that supplies the converter.

Current programmed control: summary of key points

- Output voltage can be controlled by implementing a control loop.
- Theory of chapter 1 and 2 can be applied by replacing:

$$T_u(s) \triangleq G_{PWM}(s)G_{vd}(s)H(s) \rightarrow T_u(s) \triangleq G_{vc,cpc}(s)H(s)$$

- Instead of controlling d, the new controller controls $\bar{i_c}$, this is easier because the phase of $G_{vc,cpc}(s)$ is 90° higher compared to $G_{PWM}(s)G_{vd}(s)$.
- Current programmed control is a particular case of sliding mode control, this kind of control reduces by one the order of the controlled system, this explains the "gain" of 90° when comparing to $G_{vd}(s)$.
- The above procedure can be applied to other topologies: boost, buck-boost...
- The small-signal negative resistance appears in all high efficiency converters where output power/current is controlled (nearly always the case).
- The small-signal negative resistance affects the stability of the sources powering the converter.
- Bandwidth of current programmed control is excellent yielding to unsurpassed over-current protection.

References

[1] P. M. Luca Corradini, Dragan Maksimović and R. Zane, *Digital Control of High-Frequency Switched-Mode Power Converters*.
Wiley-IEEE Press, 2015.

[2] R. W. Erickson and D. Maksimović, *Fundamentals of Power Electronics*.
Kluwer Academic Publishers, second ed., 2001.

[3] Hansruedi Bűhler, *Réglage par mode de glissement*.
Presses Polytechniques Romandes, 1986.

[4] C. W. Deisch, “Simple switching control method changes power converter into a current source,” in *1978 IEEE Power Electronics Specialists Conference*, pp. 300–306, 1978.