Lab 2 Lecture Notes

Monday, January 31, 2022 5:12 PM

Agenda:
- Inner current / outer voltage control (Background)
- week 1: controller design
- week 2: implementation and HIL
- week 3: work-time and wrap-up.

1) Background.

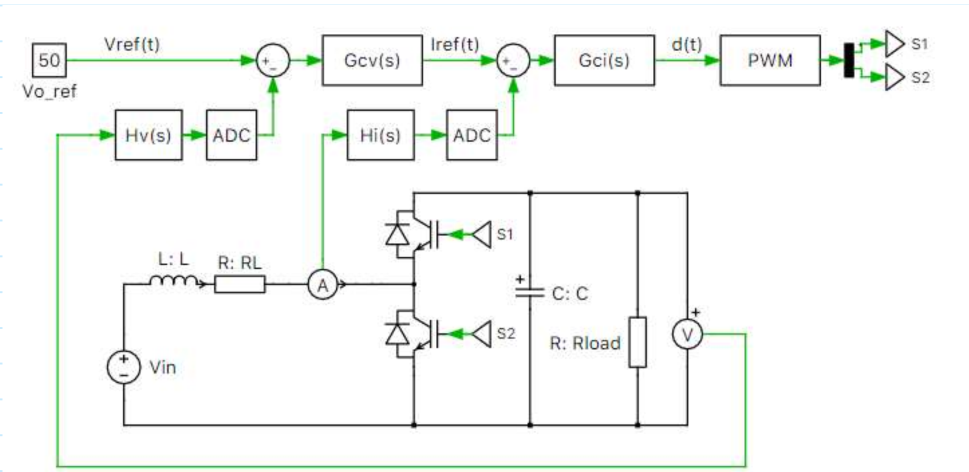


Figure 1: Conceptual diagram of ICOV control for a synchronous boost

* notice inputs / outputs

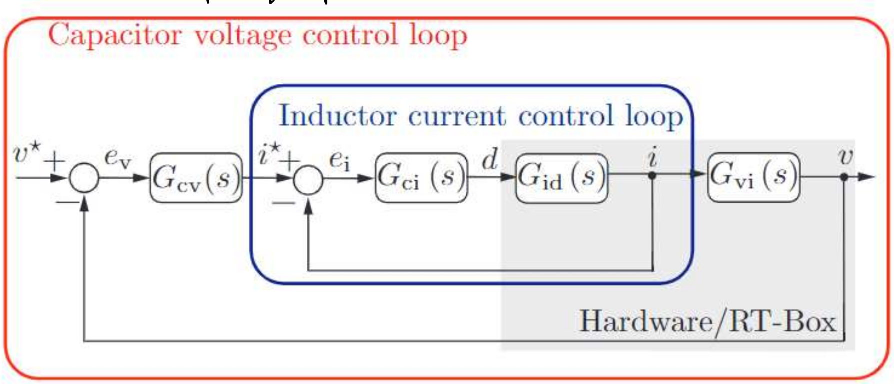


Figure 2: Transfer function diagram of ICOV control

ω_{bw} for IC loop 10x HIGHER than ω_{bw} for OV loop

Why?

- say we have OV @ 1Hz → output at highest 1Hz signal
  IC @ 10Hz any higher is filtered.

- say we have OV @ 1 Hz → output at highest 1 Hz signal
  IC @ 10 kHz   any higher is filtered.

↳ gain @ 1 Hz is high since ω_{bw} is much higher; sees close to DC.

↳ IC $\omega_{bw} \leq \frac{\omega_{sw}}{10}$

2) Controller Design: (week 1)

- similar process for IC and OV
  - specific to this problem
  - general solution discussed in lecture.

- IC controller design:

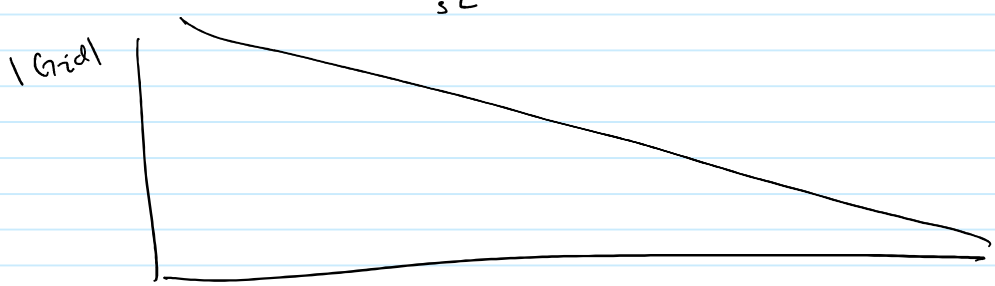
  input: i_{ref} output: d

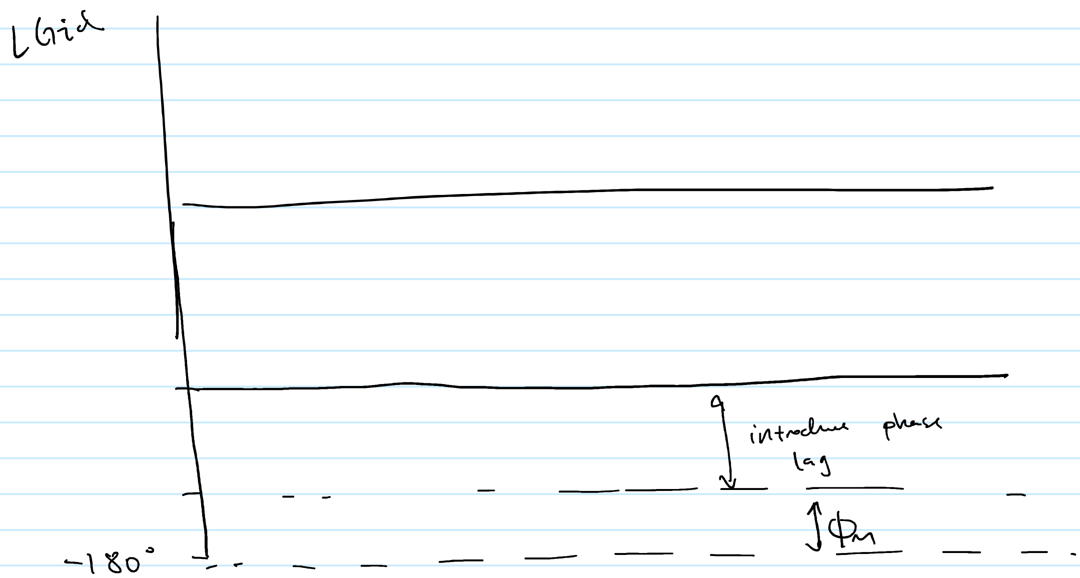
  1) select $\omega_{c,i} \leq \frac{\omega_{sw}}{10}$

     $\phi_m \leq 60°$ (I'll pick 60°)

  2) find gain / phase of uncompensated loop

  approx: $G_{id}(s) \approx \frac{V_{out}}{sL}$ at high frequency.





3.) use PI controller: $C(s) = \left(k_p + \frac{k_i}{s}\right)$

$$l_i(s) = C(s)\,P(s) = C(s)\,G_{id}(s)$$

$$\approx \left(k_p + \frac{k_i}{s}\right)\left(\frac{V_{out}}{sL}\right)$$

$$\left.\|l_i(s)\|\right|_{\omega=\omega_{bw}} = 1$$

$$\left.\angle l_i(s)\right|_{\omega=\omega_{bw}} = -180^\circ + \phi_m$$

use this to get k_p and k_i

$$\left\|\frac{V_{out}}{sL}\left(k_p + \frac{k_i}{s}\right)\right\|$$

$$\left\|\frac{V_{out}}{j\omega_c L}\left(\frac{j\omega_c k_p + k_i}{j\omega_c}\right)\right\|$$

$$\left\|\frac{V_{out}}{-L\omega_c^2}\left(j\omega_c k_p + k_i\right)\right\|$$

$$\frac{V_{out}}{L\omega_c^2}\sqrt{(k_p\omega_c)^2 + k_i^2} = 1$$

$$\frac{L\omega_c^2}{V_{out}} = \sqrt{(k_p\omega_c)^2 + k_i^2}$$

$$\boxed{\frac{L^2\omega_c^4}{V_{out}^2} = (k_p\omega_c)^2 + k_i^2} \quad (1)$$

$$\angle l_i = \angle\left(k_p + \frac{k_i}{s}\right)\frac{V_{out}}{sL} = -180^\circ + \phi_m$$

$$= \angle k_p + \frac{k_i}{s} + \angle\frac{V_{out}}{sL}$$

$$= \angle k_p + \frac{k_i}{j\omega_c} - \frac{\pi}{2}$$

$$\angle k_p + \frac{k_i j\omega_c}{-\omega_c^2} - \frac{\pi}{2}$$

$$\angle k_p - \frac{k_i j}{\omega_c} - \frac{\pi}{2}$$

$$\arctan\left(\frac{-\frac{k_i}{\omega_c}}{k_p}\right) - \frac{\pi}{2}$$

$$-\arctan\left(\frac{k_i}{\omega_c k_p}\right) - 90^\circ = -180^\circ + \phi_m$$

$$\arctan(x^{-1}) = \frac{\pi}{2} - \arctan(x)$$

$$-\left(\frac{\pi}{2} - \arctan\left(\frac{\omega_c k_p}{k_i}\right)\right) - 90^\circ = -180^\circ + \phi_m$$

$$\arctan\left(\frac{\omega_c k_p}{k_i}\right) - \cancel{180^\circ} = \cancel{-180^\circ} + \phi_m$$

$$\boxed{\arctan\left(\frac{\omega_c k_p}{k_i}\right) = \phi_m} \quad (2)$$

note: $k_p + k_i$ $G_{PI}\ (1 + \omega_{PI})$ $= G_n$

note: $k_p + \frac{k_i}{s} = G_{PI\infty}\left(1 + \frac{\omega_{PI}}{s}\right) = G_c$

Use (1) and (2) to get $G_{PI\infty}, \omega_{PI}$

- Simulate current control.
  - Starter file for control in Canvas
    - this is a transfer function model
    - also need to implement w/ switched model

- OV controller design:

  input: v_{ref} output: i_{ref}

  select $\omega_{c,v} \leq \frac{\omega_{ci}}{10}$

  $\left\| \left(k_p + \frac{k_i}{s}\right) G_{vi} \right\| = 1 \rightarrow \boxed{\sqrt{k_p^2 + \left(\frac{k_i}{\omega_c}\right)^2}\left[\frac{D'R}{2} - \frac{R_L}{2D'}\right]}$ (3)

  $\angle \left(k_p + \frac{k_i}{s}\right) G_{vi} = -180° + \phi_m$

  $\angle k_p + \frac{k_i}{s} + \angle G_{vi}$

  ↳ approx 0 at low frequency.

  $-\arctan\left(\frac{k_i}{\omega_c k_p}\right) = -180° + \phi_m$

  $\therefore \arctan\left(\frac{k_i}{\omega_c k_p}\right) \cancel{-180°} = \cancel{-180°} + \phi_m$

  $\boxed{\arctan\left(\frac{k_i}{\omega_c k_p}\right) = \phi_m}$ (4)

- use (3) (4) to get k_p / k_i for voltage controller.

3) Implementation (week 2)

- Set up RT BOX (see auxilliary video)

- scaling
  - scale in PLECS, send to MCU
  - read in ADC pin
  - scale to actual numerical value

scale: $\frac{3.3V}{\text{max value + safety}}$ → max ADC voltage

i_L: $\frac{3.3V}{} \cdot \frac{3.3V}{}$ V_{out}: $\frac{3.3V}{} \cdot \frac{3.3V}{}$

max value + safety

i_L: $\frac{3.3V}{10+2A} = \frac{3.3V}{12A}$ V_{out}: $\frac{3.3V}{48+12V} = \frac{3.3V}{60V}$

V_{in}: $\frac{3.3V}{24+6V} = \frac{3.3V}{30V}$

• using ≈ 25% SF

• you can choose these based on your design last quarter

- implement PI in code:
  - $i_{err} = i_{ref} - i$
  - $y_{ipr} = k_p i_{err}$ ↱ can also use an array.
  - $y_{i\,int} = T_{samp}(0.5)(i_{err} + i_{err\_prev}) k_i + y\_int\_prev$
    - ↳ trapezoidal integration
  - same idea for V

- saturation
  - saturate $y_{i\,int}$ [-0.95, 0.95]
  - saturate y_i on max/min duty
  - saturate y_v on max/min current

code structure: (suggested)

Globals / const defines
- state variables for at least 1 previous cycle
- reference variables
- saturation values
- controller gains
- scalings

ADC interrupt:

// voltage control

$$\left.\begin{aligned} V_{err} &= V_{ref} - V_o \\ Y_{vpr} &= V_{err} \cdot k_{p,v} \\ y_{int,v} &= (T_s)(0.5)(V_{err} + V_{err\,prev}) k_{i_v} + Y_{v_{int}}\,prev \end{aligned}\right\} \text{PI calculation}$$

if $y_{int,V}$ too high/low:
saturate.

update y_{prev}, $y_{v,int}\,prev$, $V_{err}\,prev$, etc.

$Y_V = Y_{int} + Y_{pr}$

update y_{prev}, $y_{v,int}$ prev, v_{err} prev, etc.

$y_v = y_{int} + y_{pr}$

if y_v too high/low // y_v is current ref!
saturate

// current control.

$i_{ref} = y_v$

repeat PI calculations but for IC loop

saturate y_i if needed // y_i = duty.

update PWM duty register (CMPA)