FOR ENERGY EFFICIENT INNOVATIONS

THINK ON.

www.onsemi.com

Designing Compensators for the Control of Switching Power Supplies

Christophe Basso – Technical Fellow

IEEE Senior Member

Public Information

Jan 2023 – Rev. 0.31

Agenda

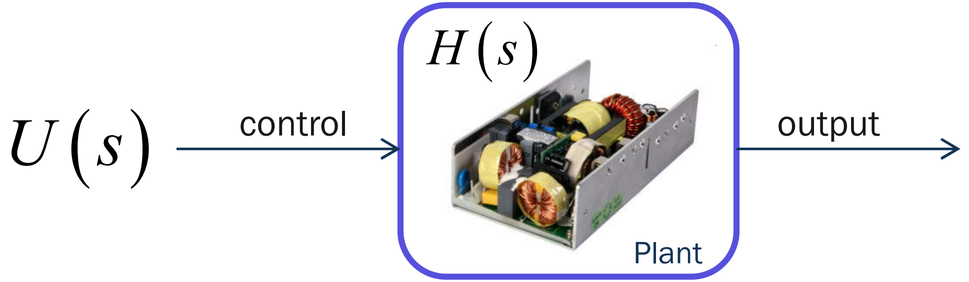
- ❑ Introduction to Control Systems
- ❑ Compensation Strategies with an Operational Amplifier
- ❑ The TL431 at Work in Compensators
- ❑ Building a PID
- ❑ Continuous-Time to Discrete-Time Domains
- ❑ Mapping and Compression
- ❑ Digital Compensation
- ❑ Practical Implementation of Filters
- ❑ Conclusion

Agenda

- ❑ Introduction to Control Systems
- ❑ Compensation Strategies with an Operational Amplifier
- ❑ The TL431 at Work in Compensators
- ❑ Building a PID
- ❑ Continuous-Time to Discrete-Time Domains
- ❑ Mapping and Compression
- ❑ Digital Compensation
- ❑ Practical Implementation of Filters
- ❑ Conclusion

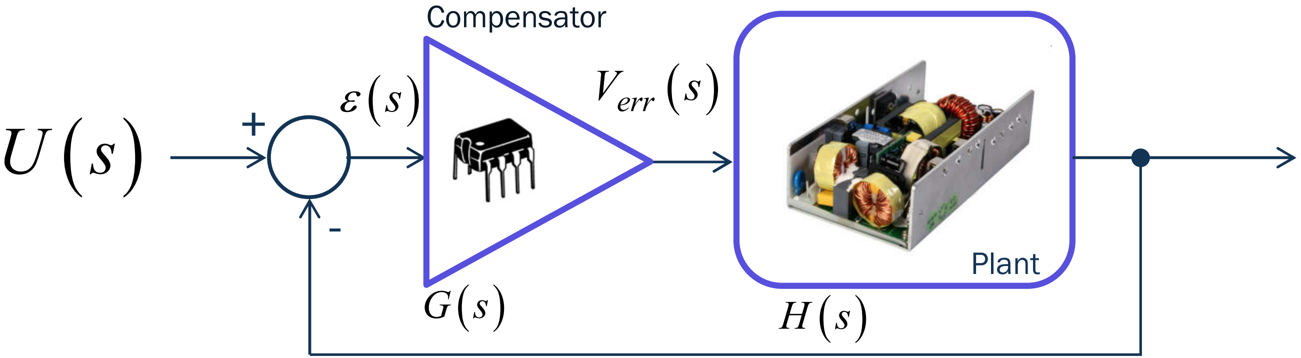
What is a Control System?

- ❑ An *open-loop* system links the output to the control variable



$$Y(s)=U(s)H(s)$$

- ❑ A *control system* observes the output and minimizes errors



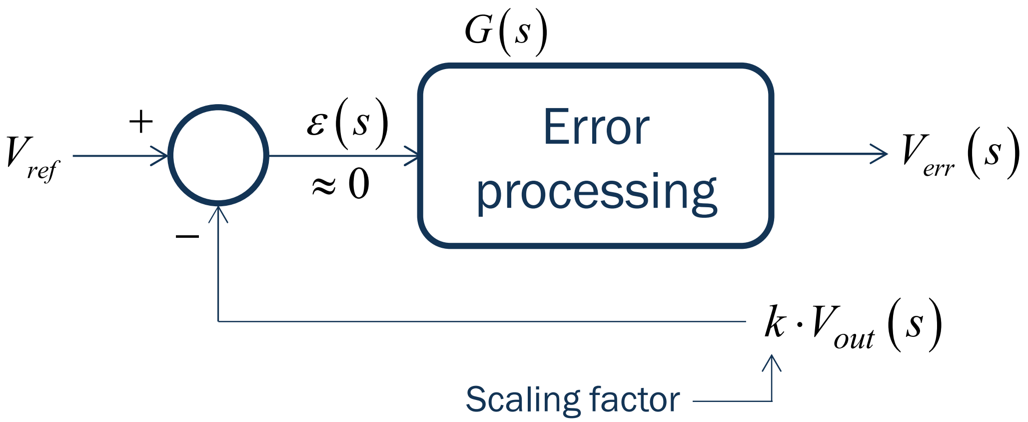
$$Y(s)=V_{err}(s)H(s)$$

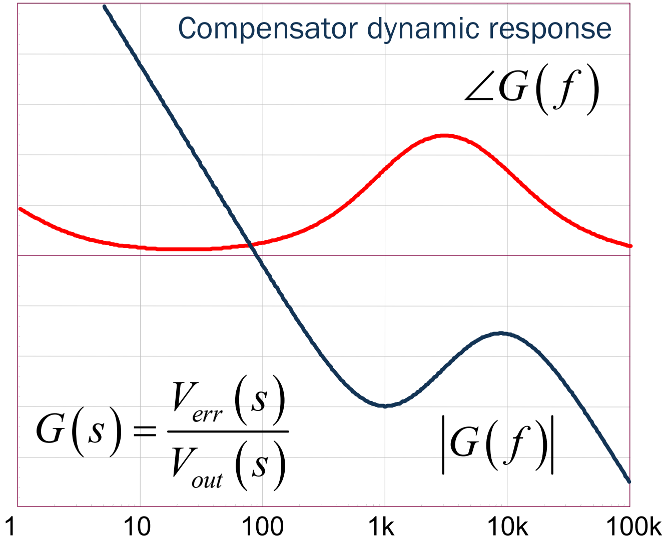
$$V_{err}(s)=\big[U(s)-Y(s)\big]G(s)$$

input output

Shaping the Loop with the Compensator

- ❑ The **compensator** builds the error variable and ensures stability
  - ➢ Insert poles and zeros to build the compensation strategy
  - ➢ Choose how to cross over at f_c with phase and gain margins

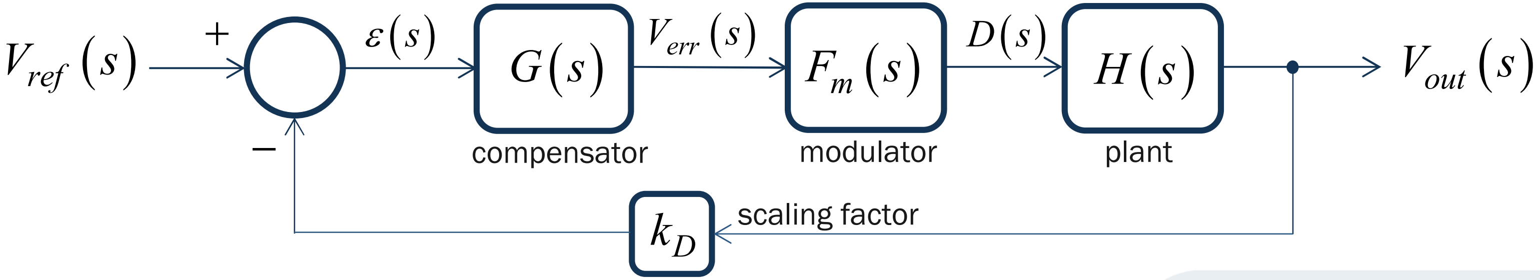




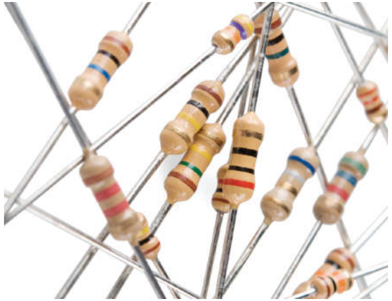
- ❑ The block amplifies and shapes the error ε between V_{ref} and V_{out}
  - ➢ Minimize the error between the setpoint and the output

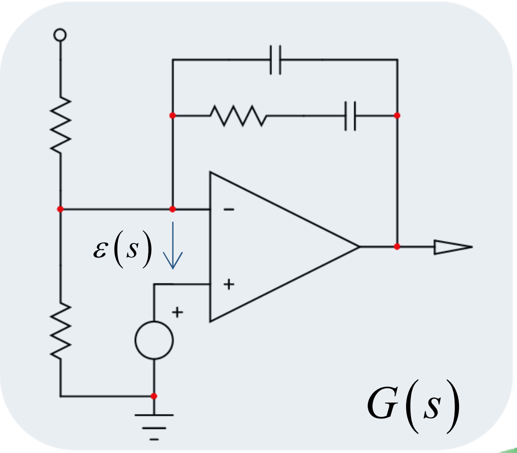
Building the Compensator the Analog Way

- Associate active and passive components to form the compensation chain



1. Select poles/zeroes placement
2. Calculate components values
3. Solder resistors and capacitors

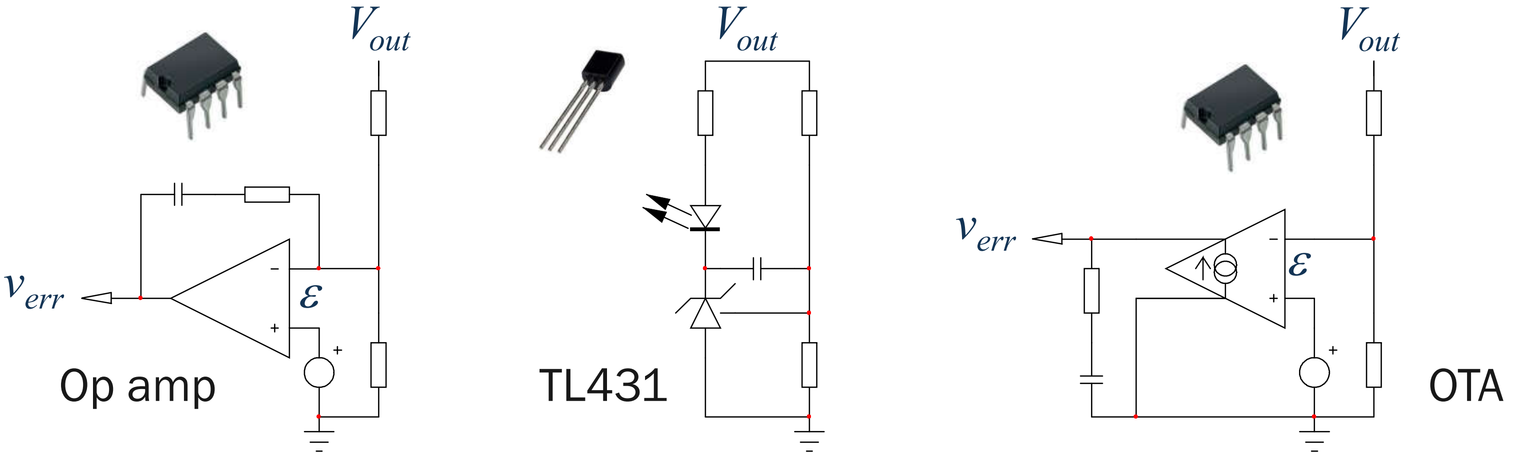




Change in strategy requires new components values

How do you Build a Compensator?

- ❑ The compensator can be implemented with analog components

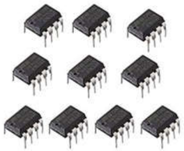


✓ Passive components suffer drawbacks:

1. Tolerance, aging
2. Sensitivity to temperature, humidity

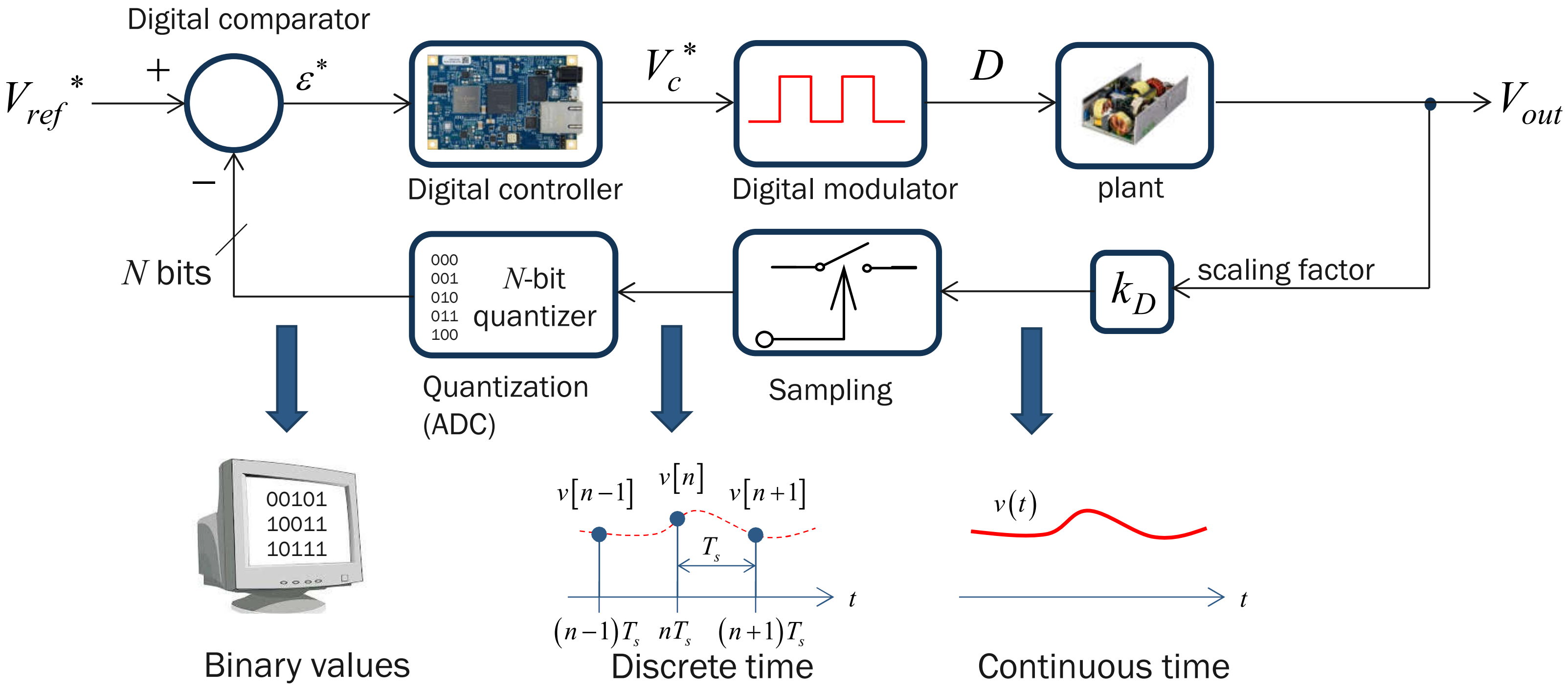
✓ Active components are not perfect!

1. Open-loop gain, bias requirements
2. Limited in bandwidth, slew-rate
3. Temperature drift



Compensating the Digital Way

- ❑ A digitally-controlled system is a discrete-time system
  - ➢ It contains continuous- and discrete-time components



* designates a discrete variable

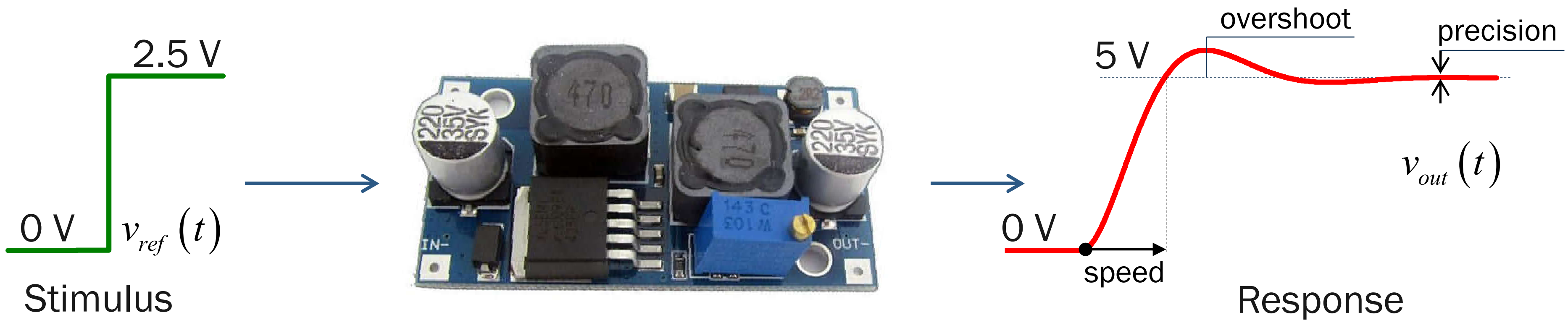
Public Information

M. Jovanović, *Introduction to Digital Control of Switch-Mode Converters*, in-house course, PHX 2014

Why do we Need to Close the Loop?

- We want to compensate the power stage deficiencies to obtain:

➢ Speed		✓ High bandwidth
➢ Precision	➡	✓ Large dc gain
➢ Robustness		✓ High gain below f_c



What compensation strategy?

Choosing the Crossover Frequency f_c

- Selecting f_c depends on the topology and its control mode

Topology	Voltage Mode	Current Mode
Buck	$3 \cdot f_0 < f_c < \frac{F_{sw}}{2}$ *	$f_c < \frac{F_{sw}}{2}$ *
Boost	$3 \cdot f_0 < f_c < 0.3 \cdot f_{RHPZ}$	$f_c < 0.3 \cdot f_{RHPZ}$
Buck-boost	$3 \cdot f_0 < f_c < 0.3 \cdot f_{RHPZ}$	$f_c < 0.3 \cdot f_{RHPZ}$

Continuous conduction mode

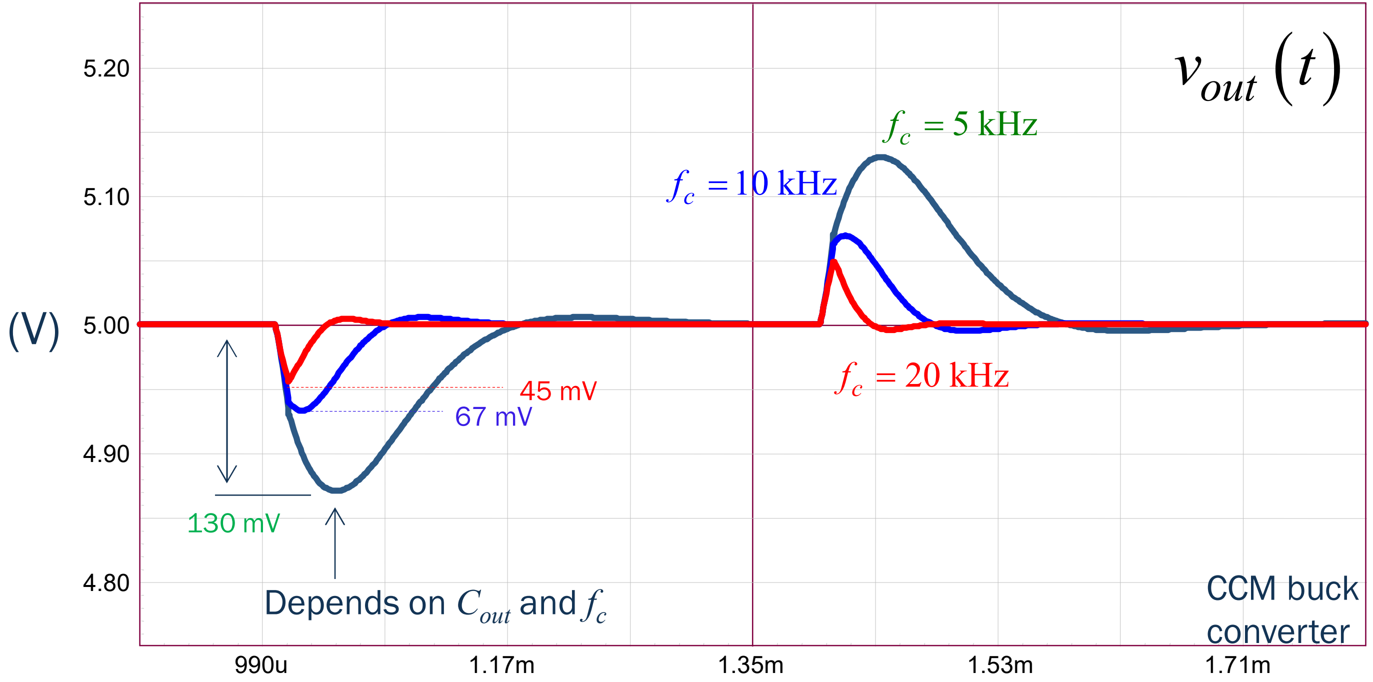
* Theoretical upper limit

Too high a crossover frequency affects susceptibility to noise!

f_0 is the LC network resonant frequency

Crossover Frequency Impacts Response Speed

- Adjust f_c to meet the transient response you want



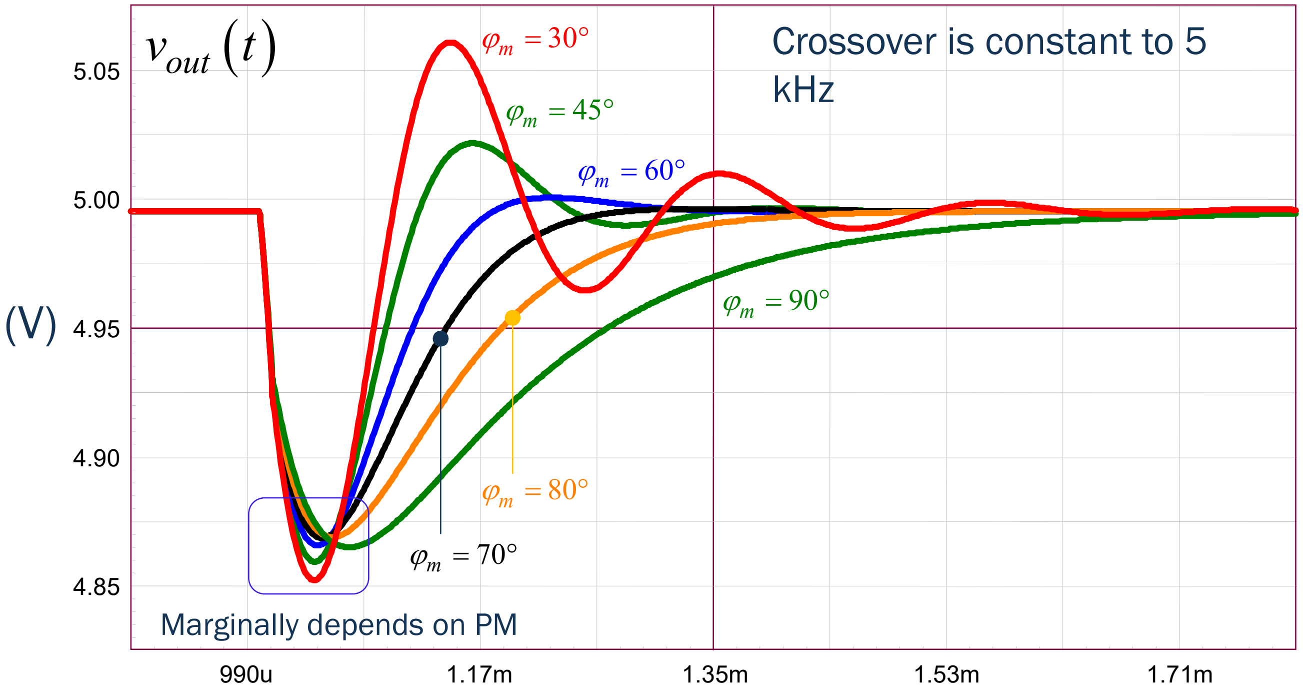
$$\Delta V_{out} \approx \frac{\Delta I_{out}}{2\pi f_c C_{out}}$$

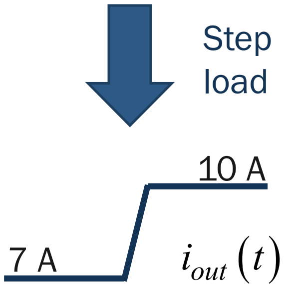
$$f_c = \frac{\Delta I_{out}}{\Delta V_{out}} \frac{1}{2\pi C_{out}}$$

$r_C << Z_{C_{out}}$ @ f_c

Transient Response Depends on Phase Margin

- Phase margin affects overshoot and recovery time





ΔI_{out} = 3 A in 3 µs

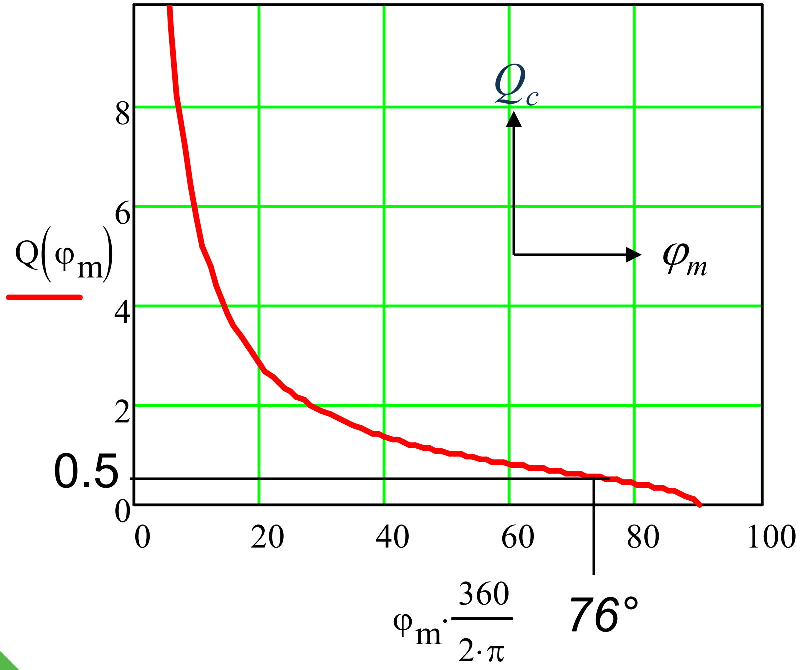
Open-Loop Phase Margin Affects Closed-Loop Response

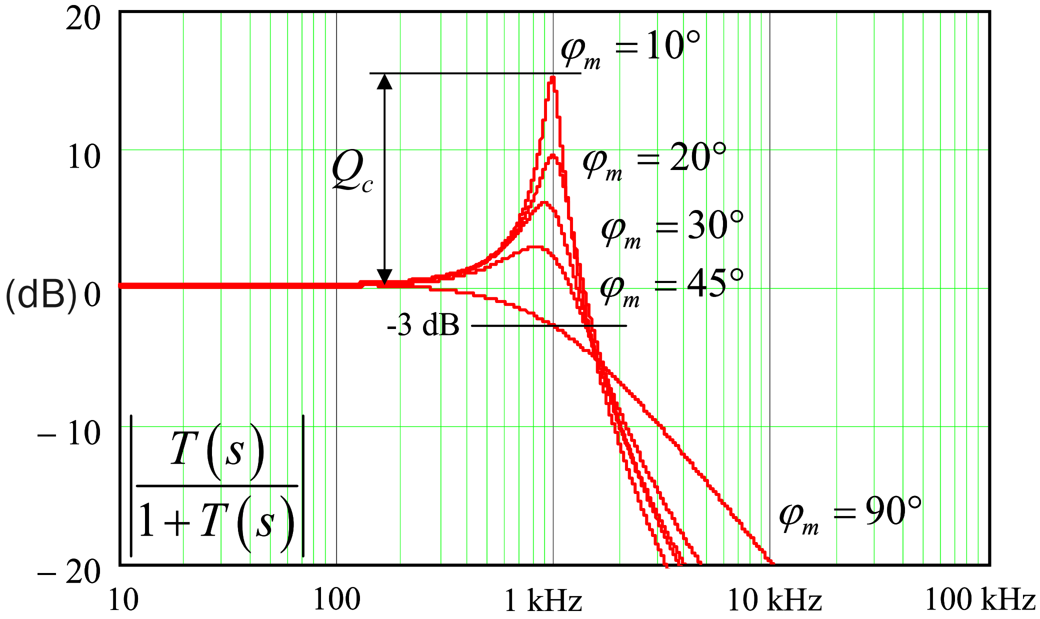
$$Q_c = \frac{\sqrt{\cos(\varphi_m)}}{\sin(\varphi_m)}$$

φ_m Open-loop phase margin

Q_c Closed-loop quality factor

It is an approximation for a 2nd-order system!





Closed-loop response

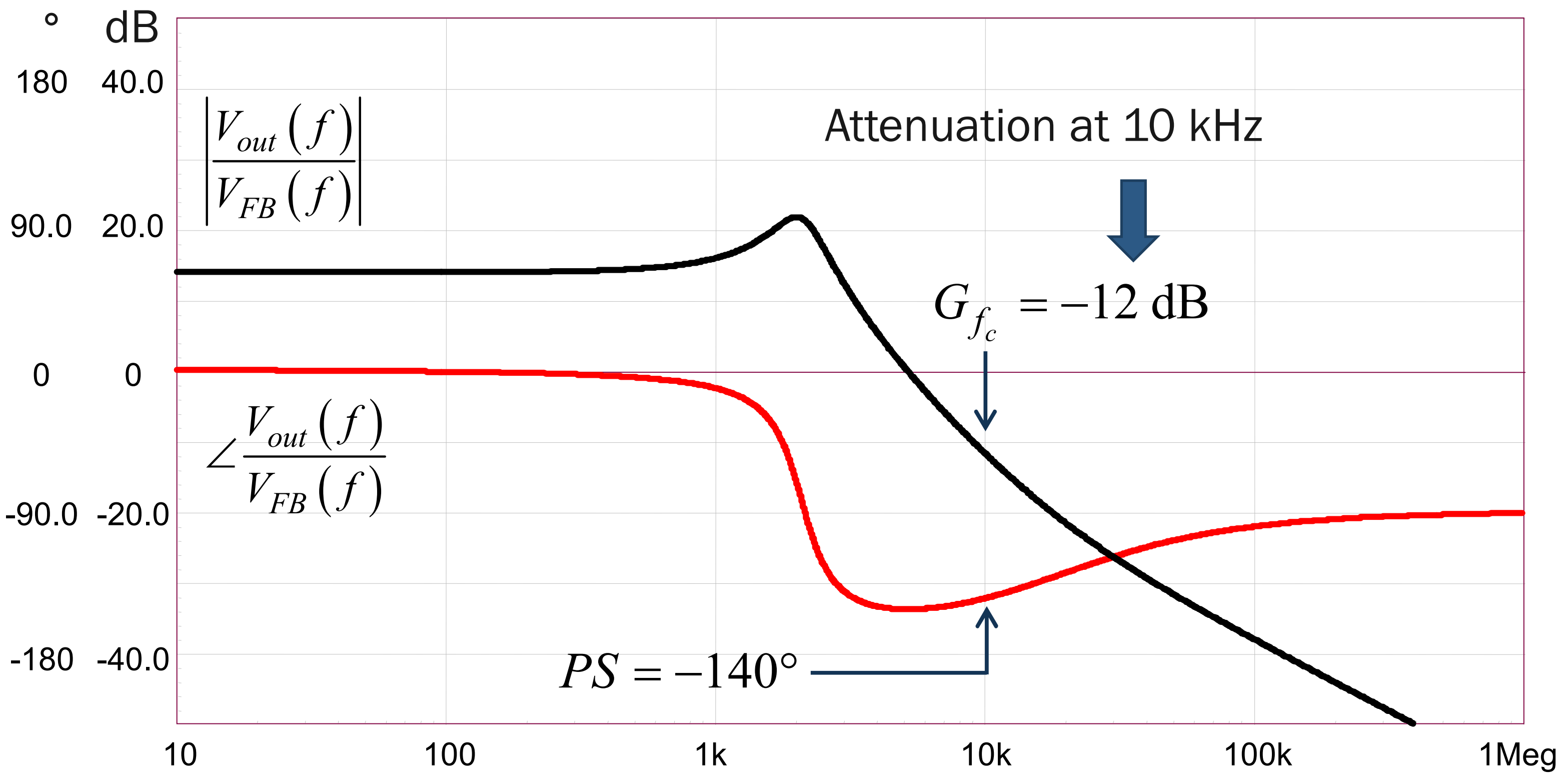
C. Basso, "The Dark Side of Loop Control Theory", APEC 2012 Professional Seminar

Agenda

- ❑ Introduction to Control Systems
- ❑ **Compensation Strategies with an Operational Amplifier**
- ❑ The TL431 at Work in Compensators
- ❑ Building a PID
- ❑ Continuous-Time to Discrete-Time Domains
- ❑ Mapping and Compression
- ❑ Digital Compensation
- ❑ Practical Implementation of Filters
- ❑ Conclusion

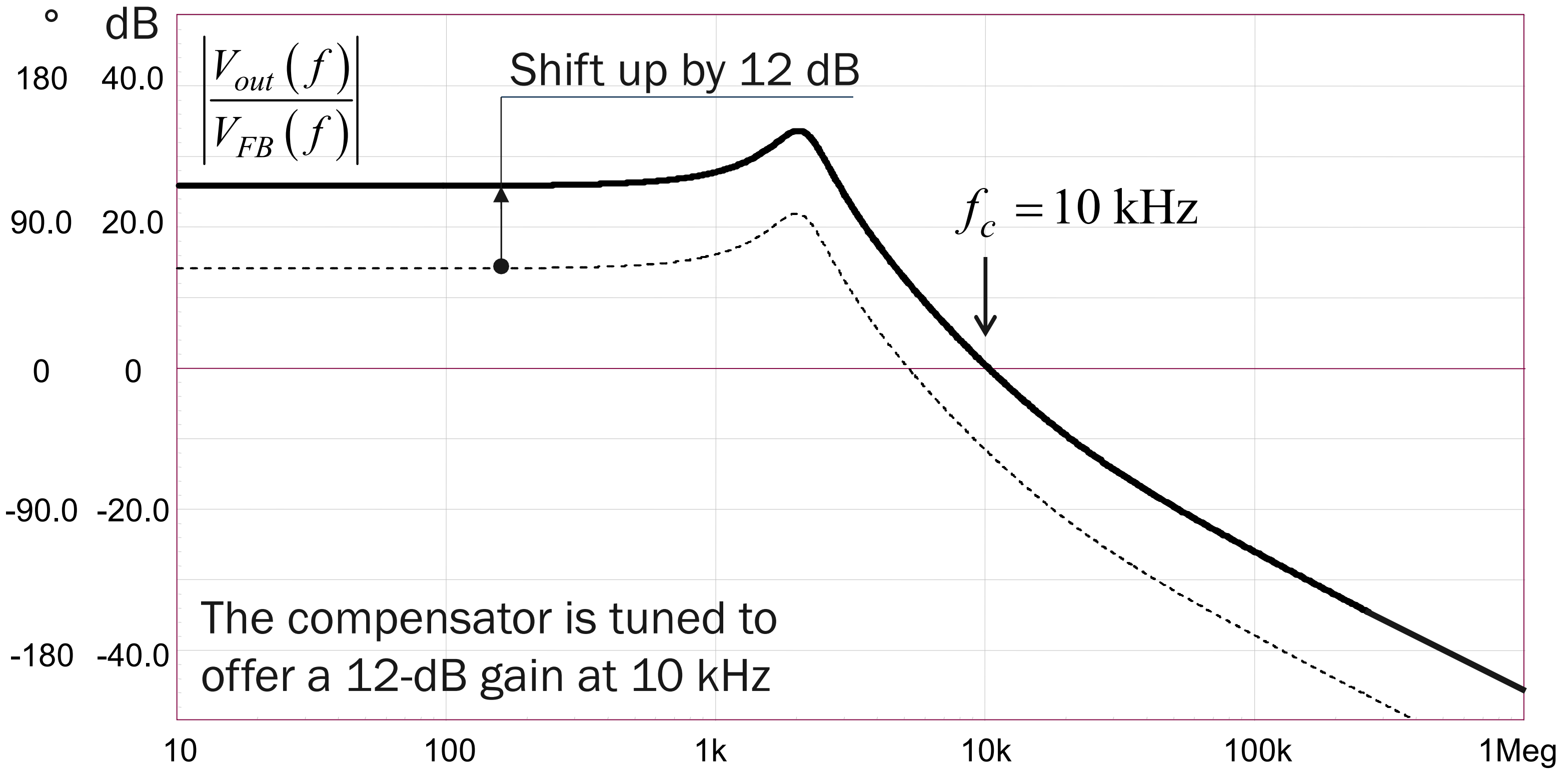
Meeting the Selected Crossover Frequency

- Extract magnitude and phase of the power stage transfer function at f_c



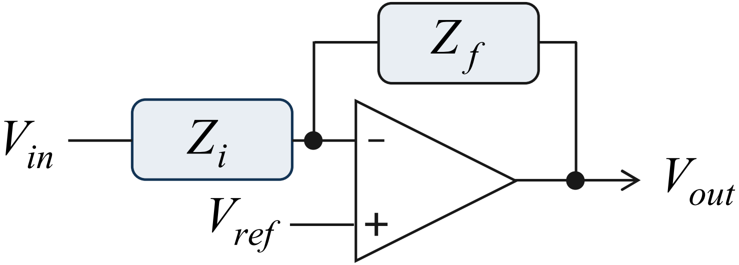
Shift the Magnitude Curve Up or Down to Meet Crossover

- Tailor the compensator to offer a 12-dB gain at the selected crossover



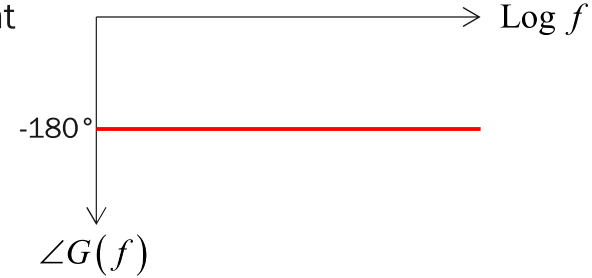
A Permanent 270° Phase Lag

- The compensator is typically built around an inverting op-amp



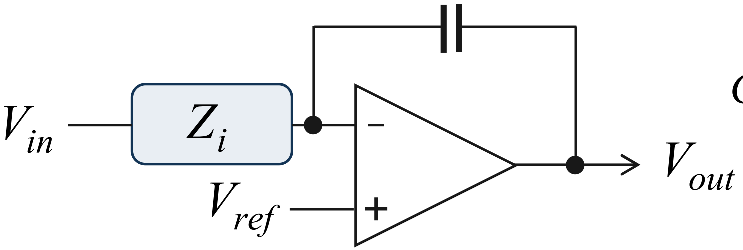
$$G(s) = -\frac{Z_f(s)}{Z_i(s)}$$

Permanent phase lag



- A compensator typically includes a pole at the origin: $|G(0)| \to \infty$

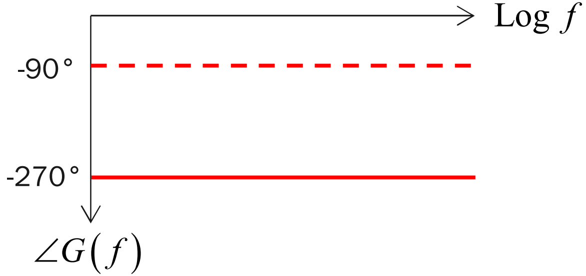
$$Z_f = \frac{1}{sC_f}$$



$$G(s) = -\frac{1}{sC_f}\frac{1}{Z_i(s)}$$

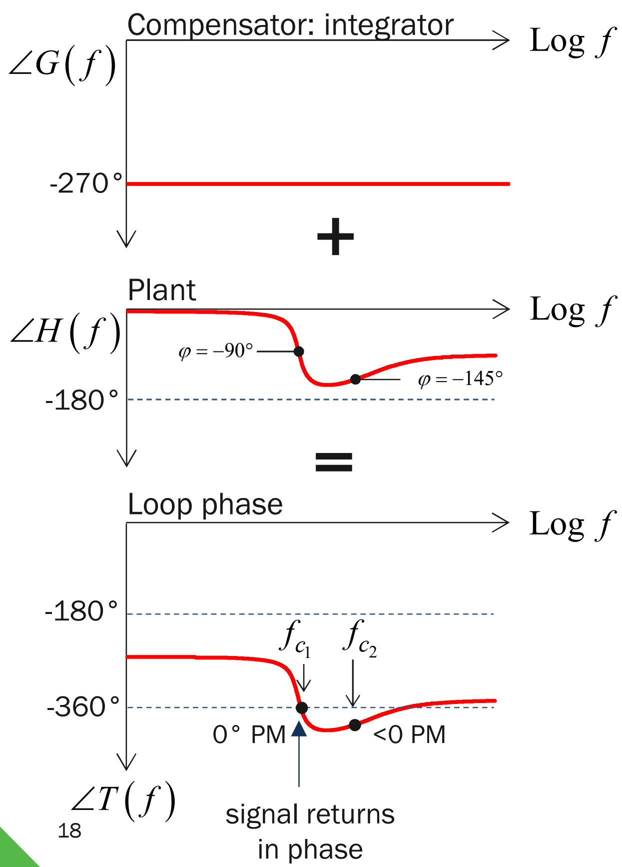
pole $s = 0$

Add another -90°



Stay Away from the 360° Limit: Build Phase Margin

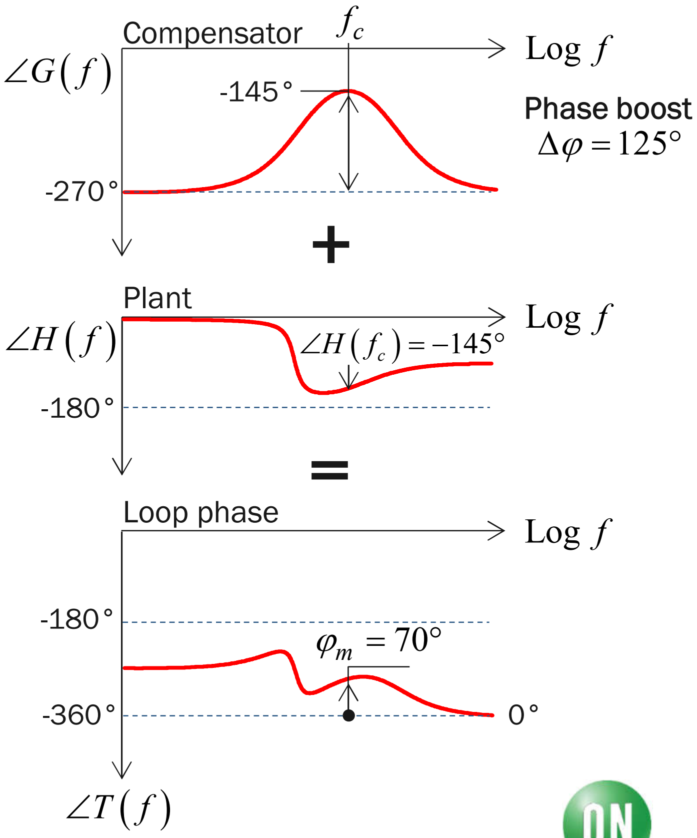
- ❑ Add up plant and compensator phase responses to total less than -360°



Decrease compensator lag near crossover

$$\angle H(f_c) - 270° + boost = -360° + \varphi_m$$

$$\boxed{boost = \varphi_m - \angle H(f_c) - 90°}$$



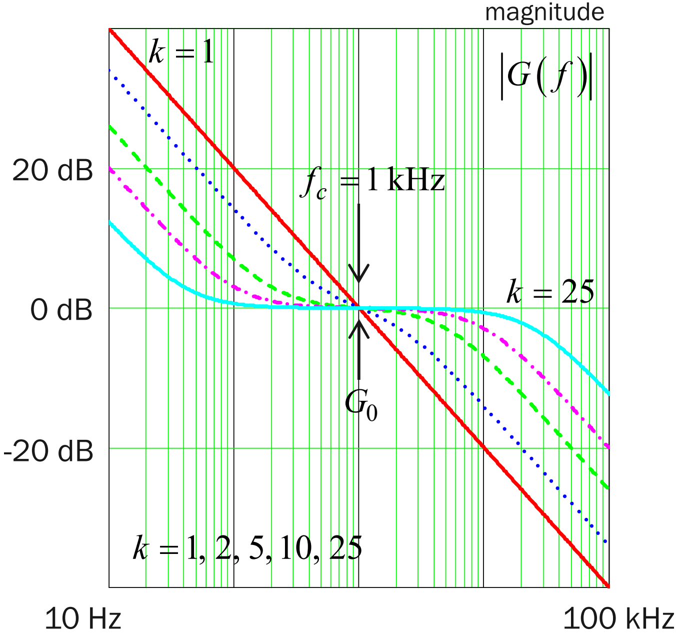
Public Information

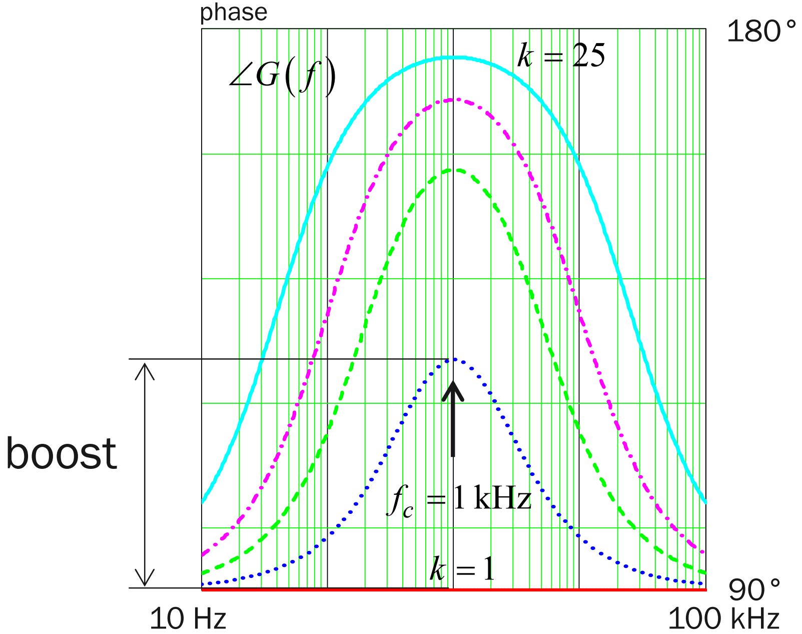
How to Create and Adjust Phase Boost

- Combining a zero and a pole lets you adjust the boost from 0° to 90°

$$G(s) = -G_0 \frac{1+\frac{\omega_z}{s}}{1+\frac{s}{\omega_p}} \qquad \angle G(f_c) = \pi - \tan^{-1}\frac{f_z}{f_c} - \tan^{-1}\frac{f_c}{f_p} \xrightarrow{\text{peaks at}} f_c = \sqrt{f_z f_p} \xrightarrow{\text{Apply } k \text{ factor}} \begin{array}{l} f_z = \dfrac{f_c}{k} \\ f_p = k \cdot f_c \end{array}$$

Mid-band gain





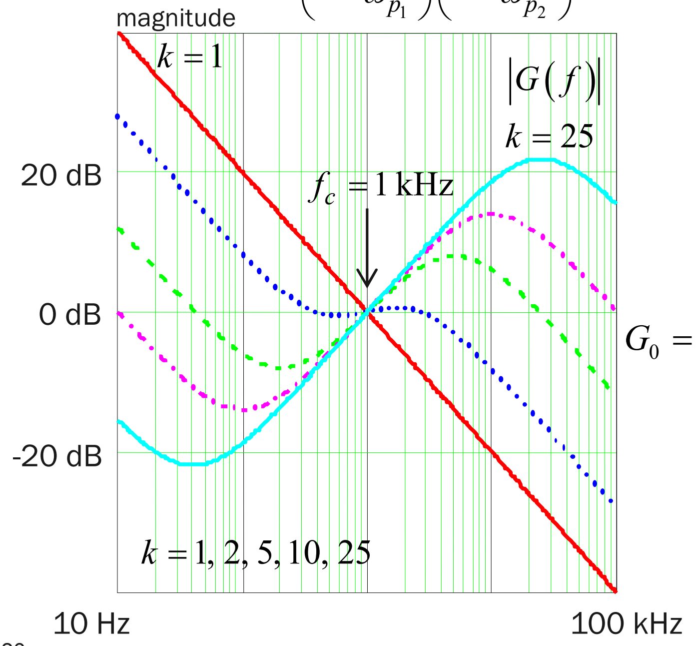
D. Venable, *The k-Factor: a New Mathematical Tool for Stability Analysis and Synthesis*, Proceedings of Powercon 10, 1983

Boosting the Phase up to 180°

- By placing a double zero and a double pole, the boost increases up to 180°

$$G(s) = -G_0 \frac{\left(1+\frac{\omega_{z_1}}{s}\right)\left(1+\frac{s}{\omega_{z_2}}\right)}{\left(1+\frac{s}{\omega_{p_1}}\right)\left(1+\frac{s}{\omega_{p_2}}\right)}$$

$$\angle G(f_c) = \pi - \tan^{-1}\frac{f_{z_1}}{f_c} + \tan^{-1}\frac{f_c}{f_{z_2}} - \tan^{-1}\frac{f_c}{f_{p_1}} - \tan^{-1}\frac{f_c}{f_{p_2}}$$

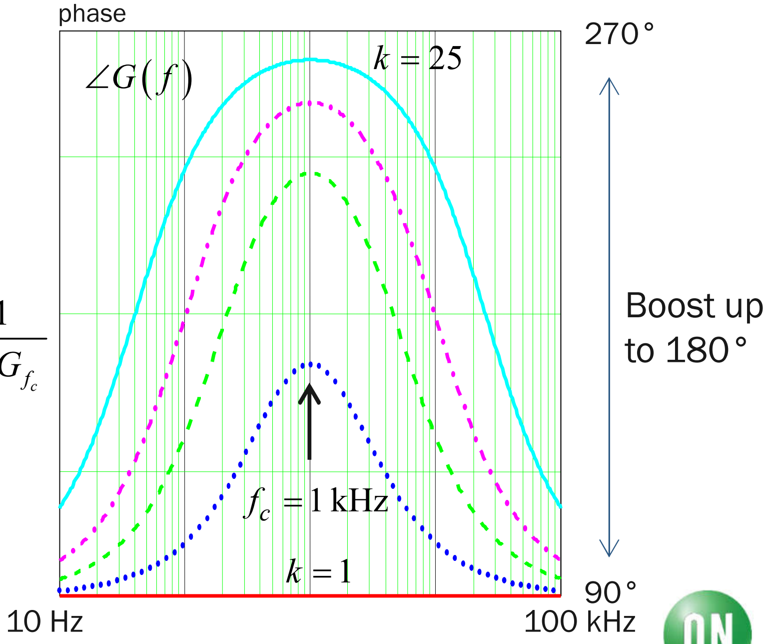


$$f_{z_1} = f_{z_2}$$

$$f_{p_1} = f_{p_2}$$

$$\downarrow$$

$$G_0 = \frac{\sqrt{1+k^2}}{G_{f_c} k^2 \sqrt{\frac{1}{k^2}+1}} = \frac{1}{k \cdot G_{f_c}}$$



There are Three Compensator Types

$$G(s) = -\frac{1}{\frac{s}{\omega_{po}}}$$

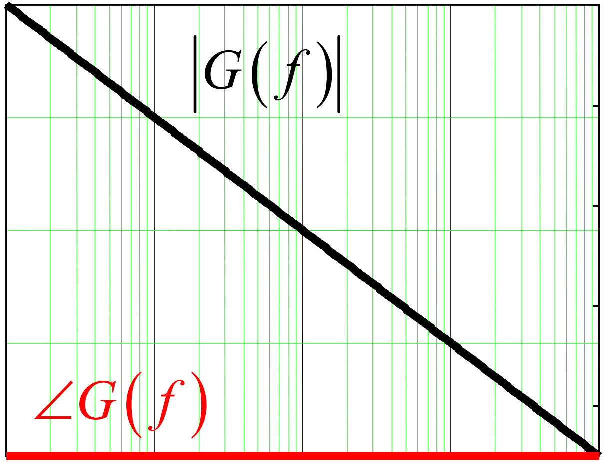
Gain

$$G(s) = -G_0 \frac{1+\boxed{\frac{\omega_z}{s}}}{1+\frac{s}{\omega_p}}$$

Inverted zero

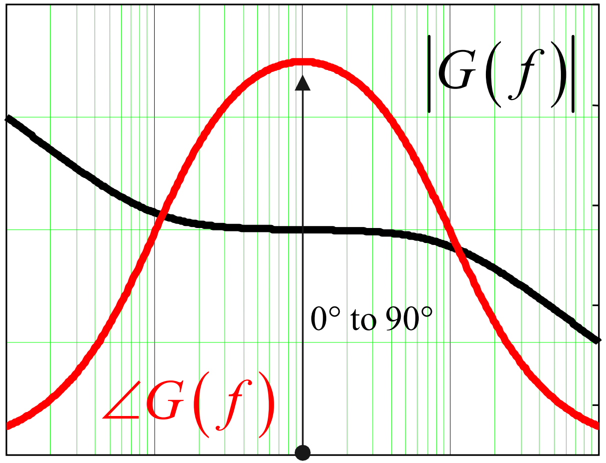
$$G(s) = -G_0 \frac{\left(1+\boxed{\frac{\omega_{z_1}}{s}}\right)\left(1+\frac{s}{\omega_{z_2}}\right)}{\left(1+\frac{s}{\omega_{p_1}}\right)\left(1+\frac{s}{\omega_{p_2}}\right)}$$

Type 1



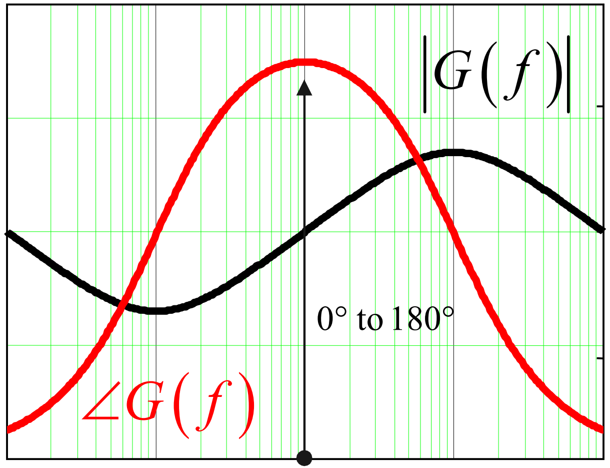
0° phase boost

Type 2



Up to 90° phase boost

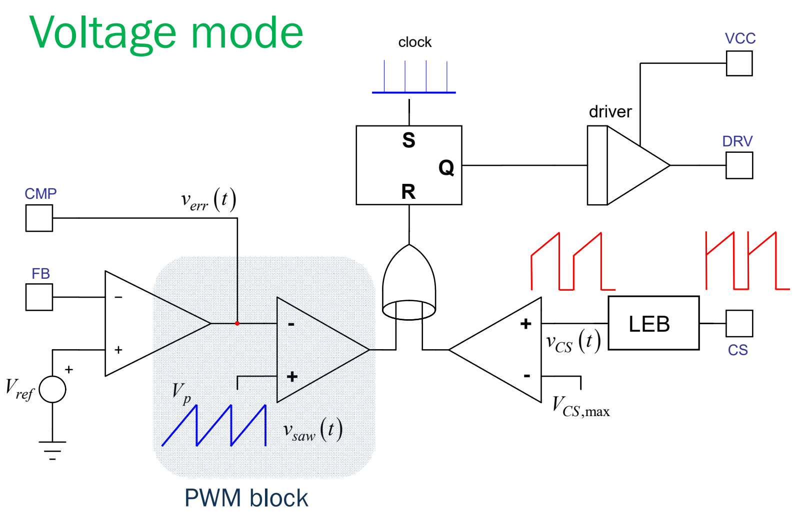
Type 3

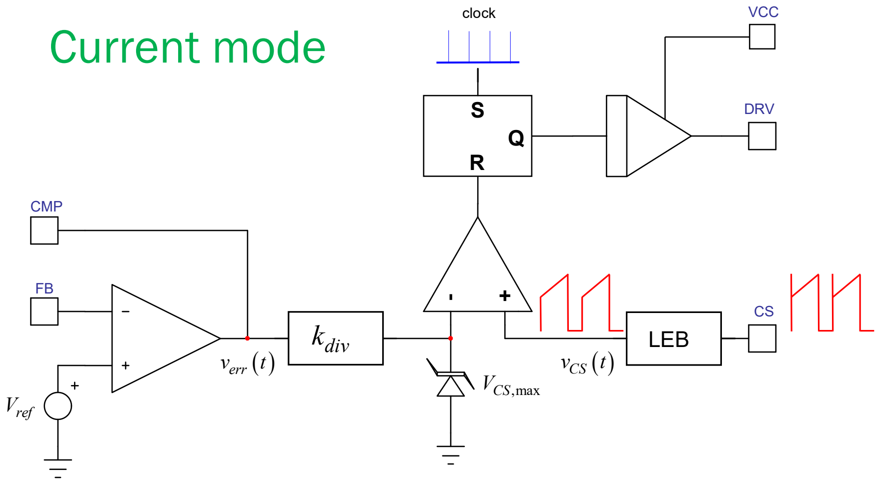


Up to 180° phase boost

Switching Topologies and Compensators

- ❑ Select the adequate compensator based on the plant phase lag





- ➢ in CCM, 2nd-order response: type 3
- ➢ in DCM, 1st-order response: type 2

Heavily-damped 2nd-order

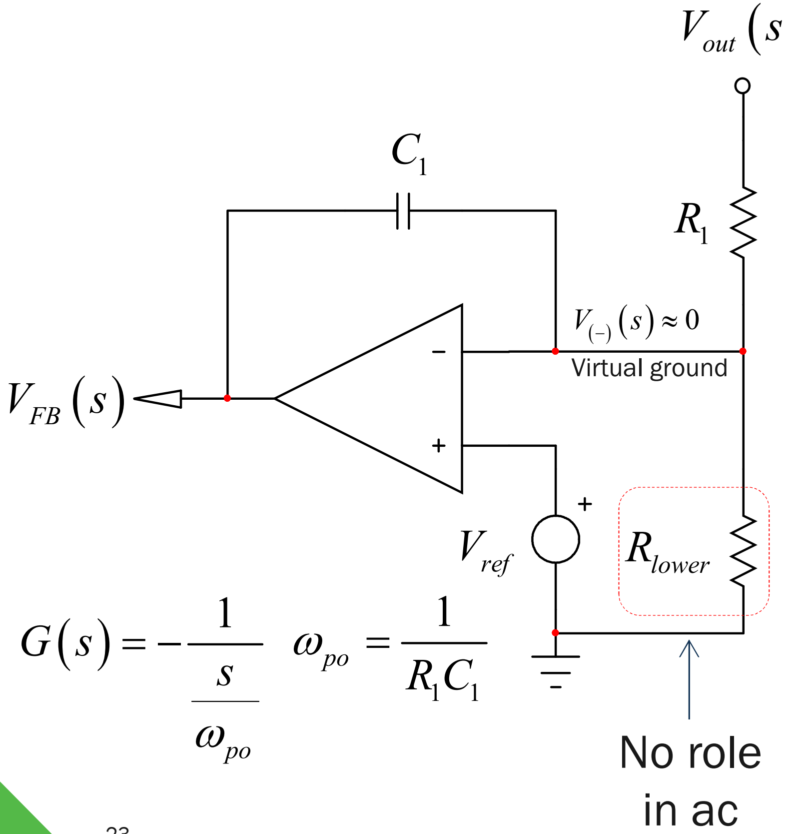
- ➢ in CCM, 3rd-order response: type 2
- ➢ in DCM, 1st-order response: type 2

Provide Gain or Attenuation at f_c: no Boost

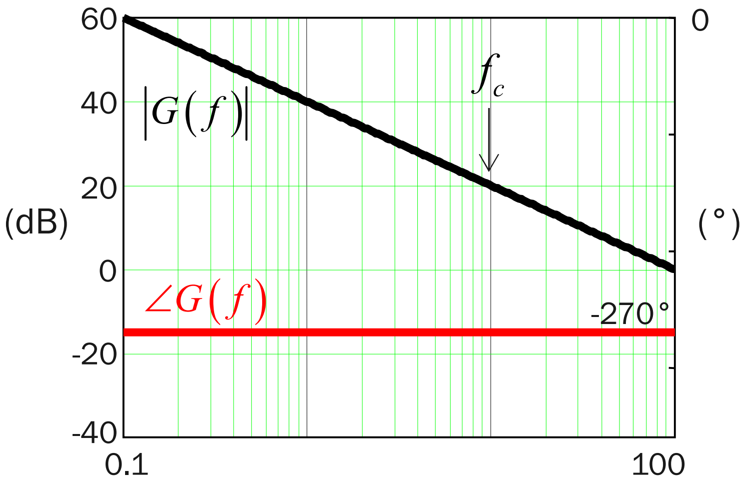
- Type 1 is a simple integrator without phase boost

$$G_{f_c} = -20\text{ dB} \ @\ f_c = 10\text{ Hz} \longrightarrow G_{f_c} = 10^{-\frac{20}{20}} = 0.1$$

$$f_{po} = \frac{f_c}{G_{f_c}} = 100\text{ Hz}$$



$$G(s) = -\frac{1}{\frac{s}{\omega_{po}}} \qquad \omega_{po} = \frac{1}{R_1 C_1}$$

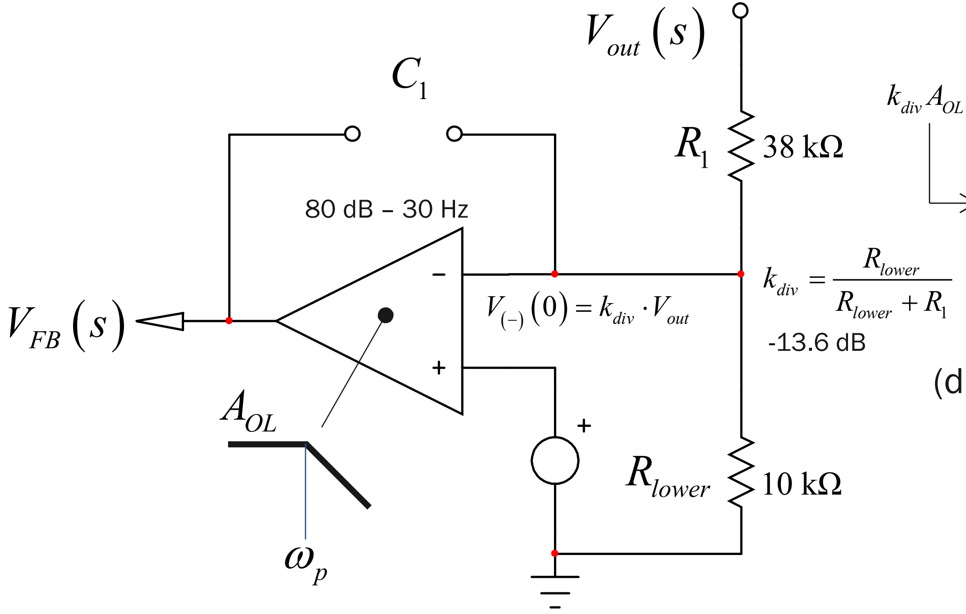


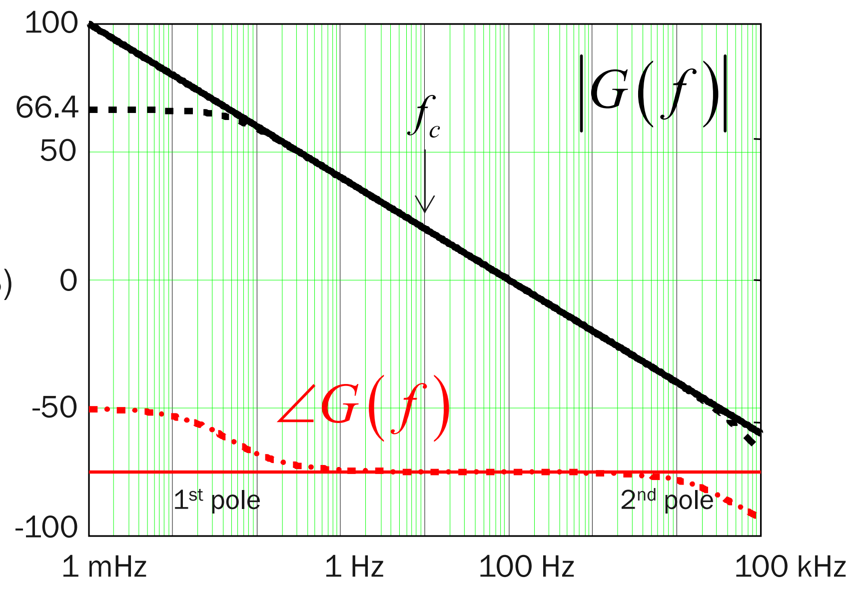
The Finite Open-Loop Gain Limits the dc Gain

- ❑ The op-amp characteristics affect the frequency response

✓ Virtual ground is lost in dc

$$G(s) \approx -\frac{A_{OL}k_{div}}{\left(1+\frac{s\cdot A_{OL}k_{div}}{\omega_{po}}\right)\left(1+\frac{s}{\omega_p A_{OL}}\right)}$$

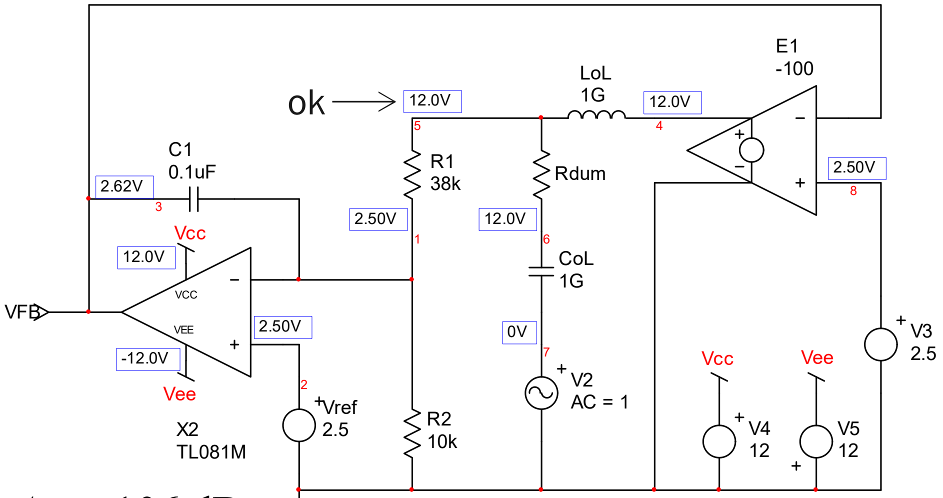




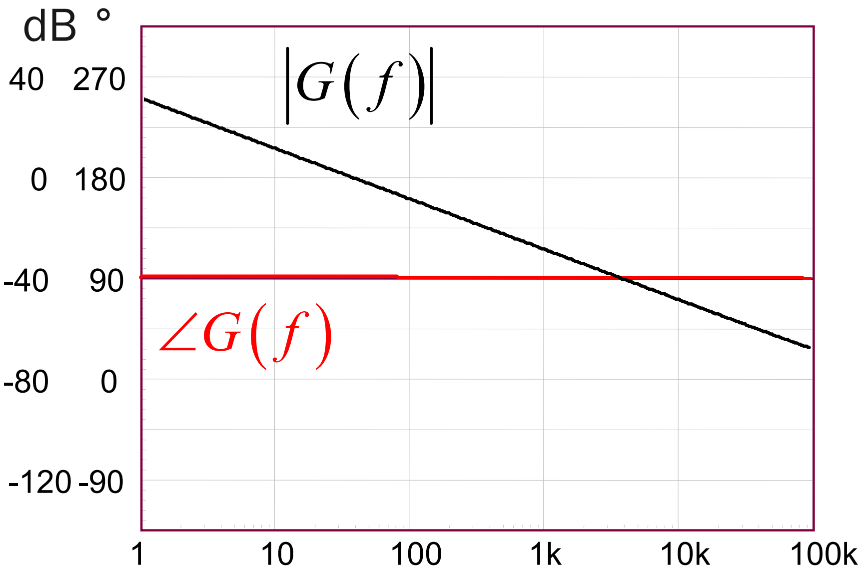
➢ Voltage divider enters the picture at dc

Simulating Large-Open-Loop-Gain Compensators?

- ❑ If you use an op-amp model, use another E-source to fix the operating point



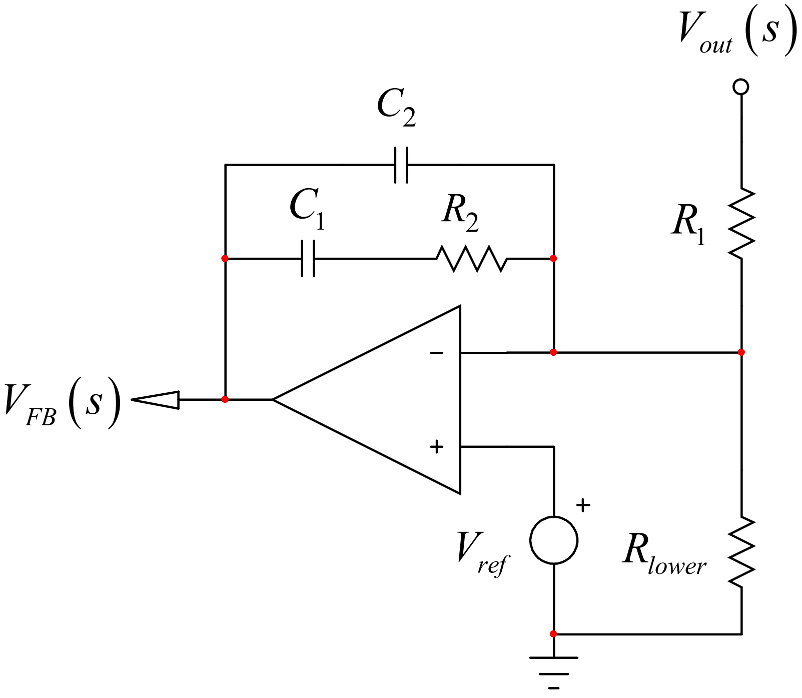
$A_{OL} = 106\text{ dB}$



- ❑ E_1 fixes the operating point to 2.5 V
- ➢ 12 V at R_1 confirms regulation is ok

Boosting the Phase at Crossover with Type 2

- By adding another zero and pole to the type 1, phase boost is created



$$G(s) = -G_0 \frac{1+\frac{\omega_z}{s}}{1+\frac{s}{\omega_p}} \qquad G_0 = \frac{R_2}{R_1}\frac{C_1}{C_1+C_2}$$

$$\omega_p = \frac{1}{R_2 \frac{C_1 C_2}{C_1+C_2}} \qquad \omega_z = \frac{1}{R_2 C_1}$$

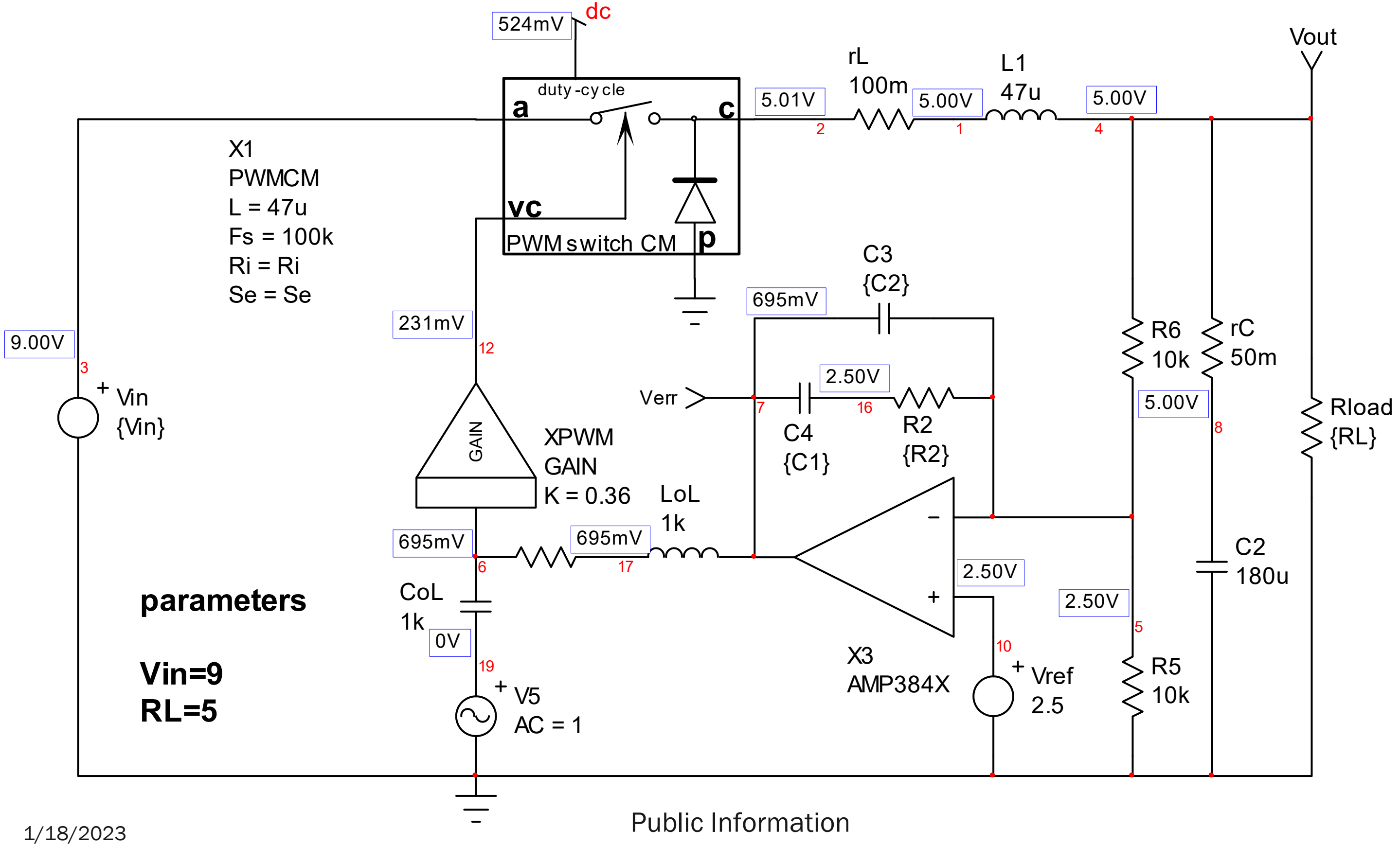
$$R_2 = \frac{R_1 f_p}{G_{f_c}\left(f_p - f_z\right)} \frac{\sqrt{\left(\frac{f_c}{f_p}\right)^2+1}}{\sqrt{\left(\frac{f_z}{f_c}\right)^2+1}}$$

$$C_1 = \frac{1}{2\pi R_2 f_z} \qquad C_2 = \frac{C_1}{2\pi f_p C_1 R_2 - 1}$$

C. Basso, *Designing Control Loops for Linear and Switching Power Supplies*, Artech House, 2012

Design Example with a Type 2 Compensator

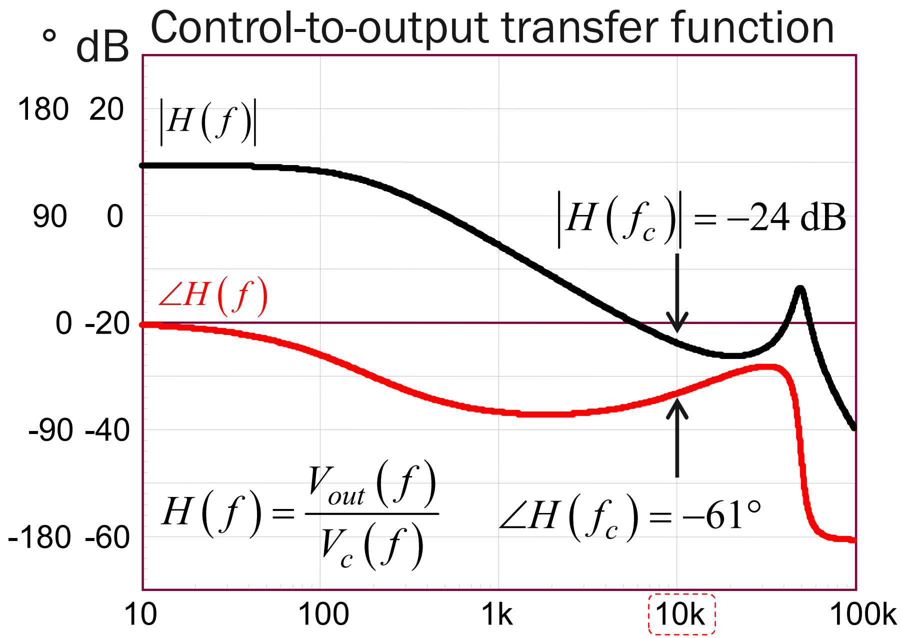
- Stabilizing a buck converter operated in current-mode control



C. Basso, Switch-Mode Power Supplies: SPICE Simulations and Practical Design, McGraw-Hill, 2014

Plot the Control-to-Output Transfer Function

- ❑ Read the Bode plot and extract the plant attenuation and lag at crossover



parameters

Rupper=10k
fc=10k
Gfc=-24
pfc=-61
pm=70
Se=20k ⟵ Slope comp.
Ri=600m

boost=pm-(pfc)-90

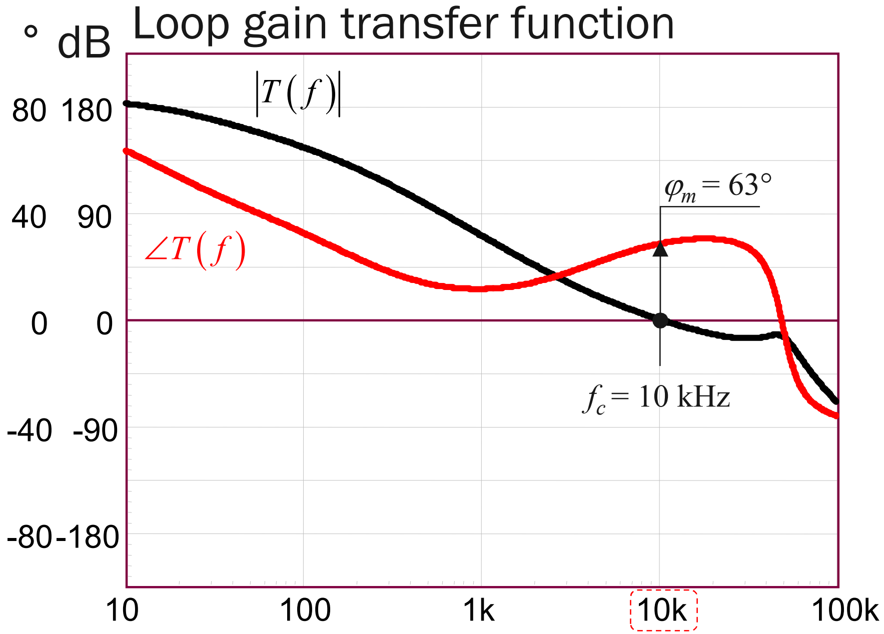
G=10^(-Gfc/20)
boost=pm-(pfc)-90
pi=3.14159
K=tan((boost/2+45)*pi/180)
C2=1/(2*pi*fc*G*k*Rupper)
C1=C2*(K^2-1)
R2=k/(2*pi*fc*C1)

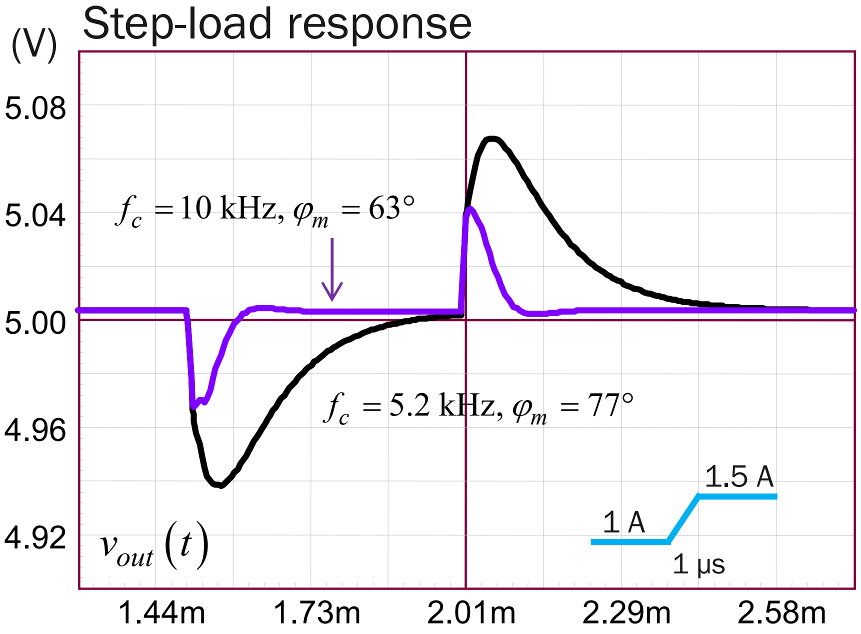
BOOST = 4.10e+001
G = 1.58e+001
PI = 3.14e+000
K = 2.19e+000
C_2 = 4.58e-011
C_1 = 1.75e-010
R_2 = 2.00e+005

Automate the calculation for easily trying different strategies

Immediately Check the Resulting Performance

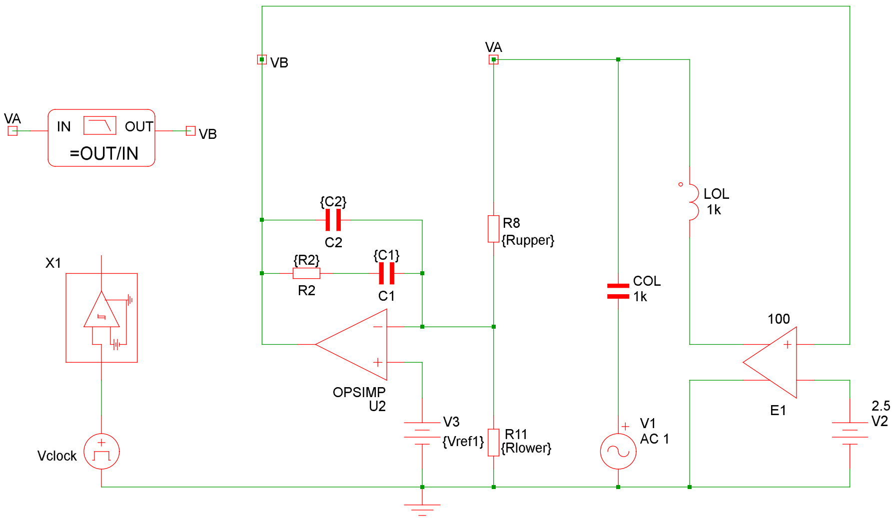
- It is easy to verify the strategy impact on the transient response



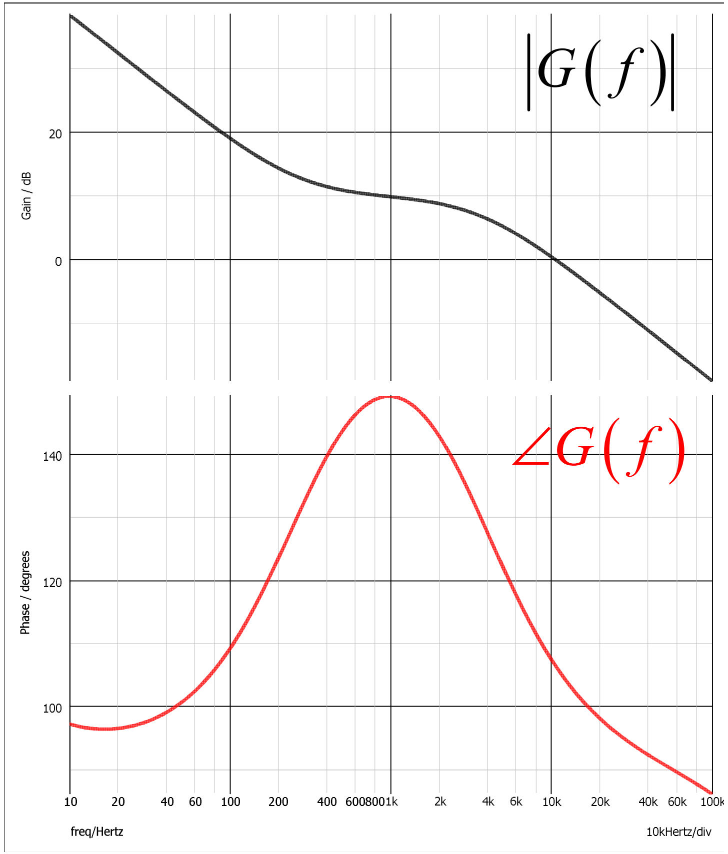


- Perform parametric sweeps with Monte Carlo analysis to check robustness

Automated Type 2 Calculations with SIMPLIS®



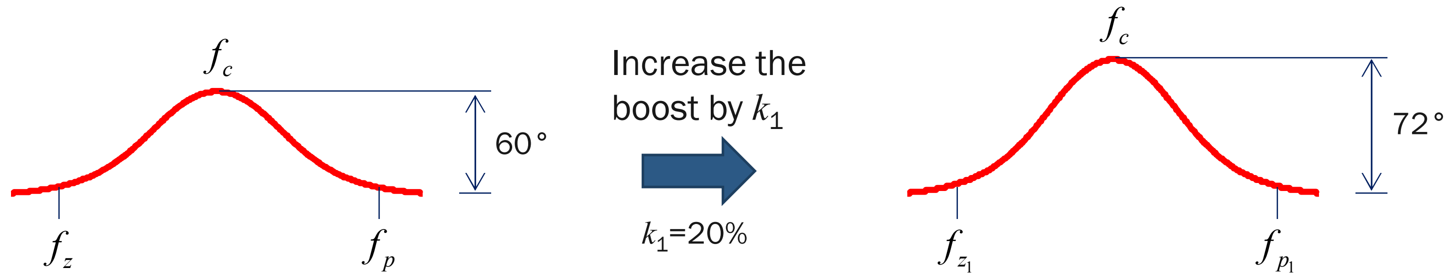

```
*
* Do not edit the below lines *
.GLOBALVAR boost=PM-PS-90
.GLOBALVAR G=10^(-Gfc/20)
.GLOBALVAR fp=(tan(boost*pi/180)+sqrt((tan(boost*pi/180))^2+1))*fc
.GLOBALVAR fz=fc^2/fp
.GLOBALVAR a=sqrt((fc^2/fp^2)+1)
.GLOBALVAR b=sqrt((fz^2/fc^2)+1)
.GLOBALVAR R2=((a/b)*G*Rupper*fp)/(fp-fz)
.GLOBALVAR C1=1/(2*pi*R2*fz)
.GLOBALVAR C2=C1/(C1*R2*2*pi*fp-1)
*
* Choose op amp characteristics *
*
.GLOBALVAR AOL=90 * open-loop gain in dB *
.GLOBALVAR POLE=30 * low-frequency pole *
.GLOBALVAR VHIGH=5 * upper output level *
.GLOBALVAR VLOW=100m * lower output level *
*
```



Press F11

Changing the Phase Boost on the Fly

- How to keep magnitude at f_c but increase the boost by a certain amount?



$$\tan^{-1}\left(\frac{f_c}{\alpha f_z}\right) - \tan^{-1}\left(\frac{f_c}{\frac{f_p}{\alpha}}\right) = k_1 \cdot boost$$

$$\tan^{-1}(x) - \tan^{-1}(y) = \tan^{-1}\left(\frac{x-y}{1+xy}\right)$$

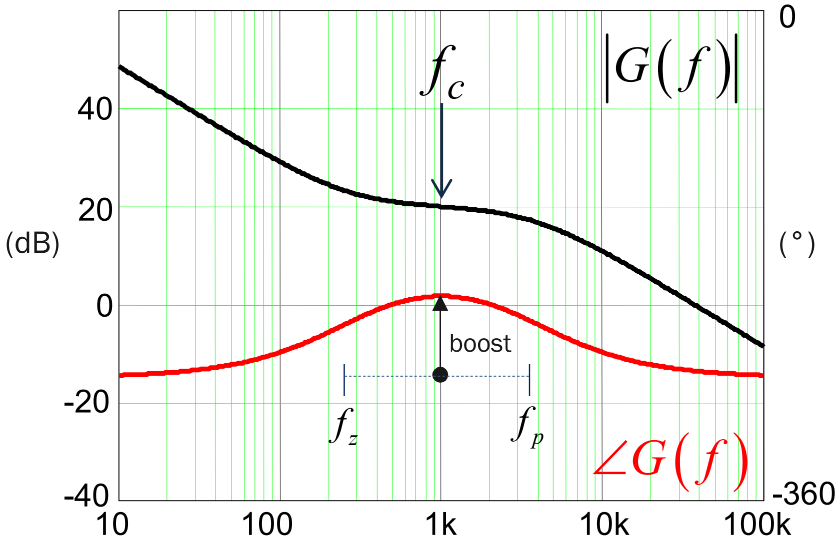
$$\alpha = \frac{\tan(boost \cdot k_1)}{2 f_z}\left[\sqrt{f_c^{\,2} + 2 f_p f_z + \frac{4 f_p f_z}{\tan(boost \cdot k_1)^2} + \left(\frac{f_p f_z}{f_c}\right)^2} - \frac{f_p f_z}{f_c} - f_c\right]$$

$$f_{z_1} = \alpha f_z$$

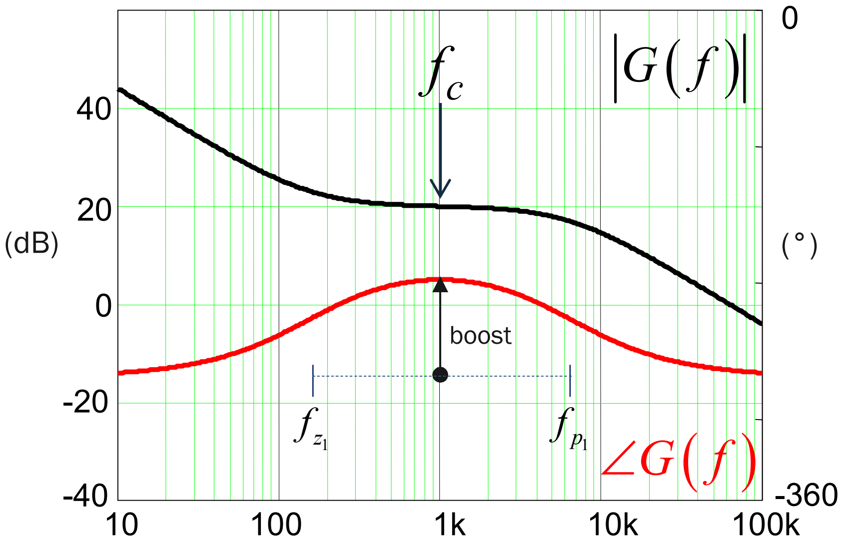
$$f_{p_1} = \frac{f_p}{\alpha}$$

Increase the Boost by 20%

- Spread pole and zero while keeping mid-band gain untouched



k_1=20%



$boost = 60°$

$f_z = 268\text{ Hz}$

$f_p = 3.73\text{ kHz}$

$\alpha = 0.5911$

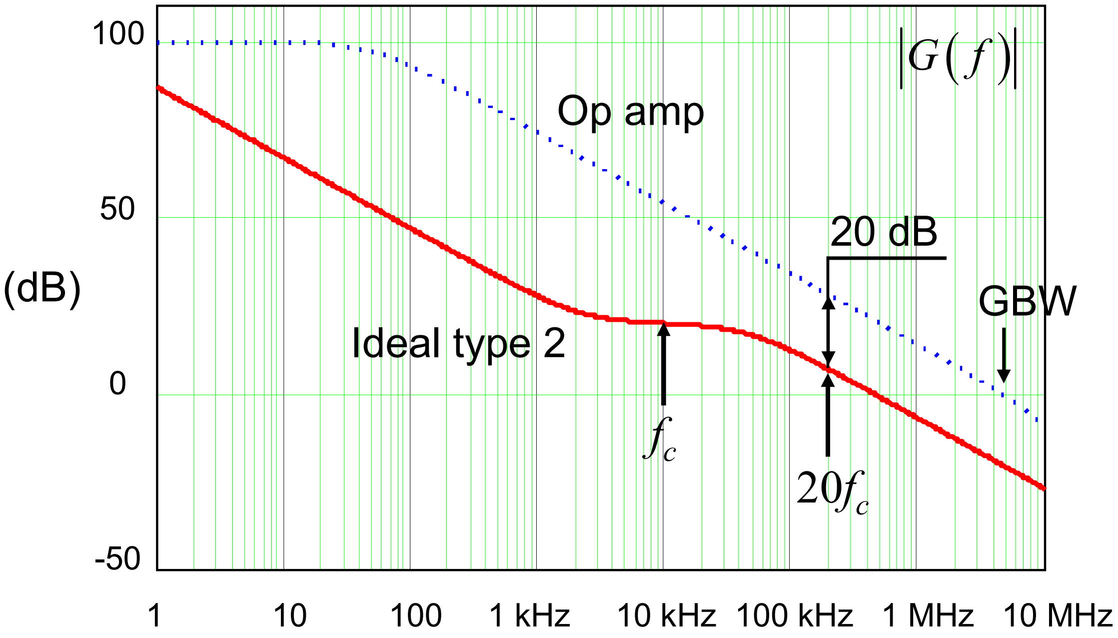
$boost = 72°$

$f_{z_1} = 158\text{ Hz}$

$f_{p_1} = 6.3\text{ kHz}$

Watch for Boost Distortion Brought by the Op-Amp

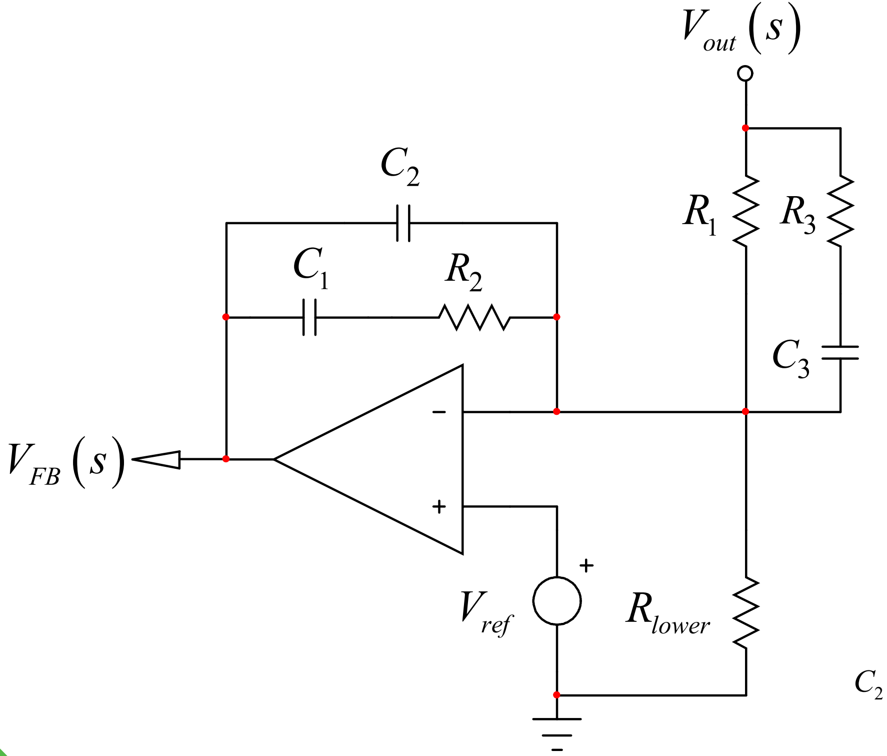
- The response must be imposed by the pole and the zero, not by the op-amp!



- Limit phase/gain distortion by selecting an op-amp still having gain at $20 \cdot f_c$

Increasing Phase Boost up to 180°

- The type 3 compensator adds another pole/zero pair to the type 2



$$G(s) = -G_0 \frac{\left(1+\frac{\omega_{z_1}}{s}\right)\left(1+\frac{s}{\omega_{z_2}}\right)}{\left(1+\frac{s}{\omega_{p_1}}\right)\left(1+\frac{s}{\omega_{p_2}}\right)} \qquad G_0 = \frac{R_2}{R_1}\frac{C_1}{C_1+C_2}$$

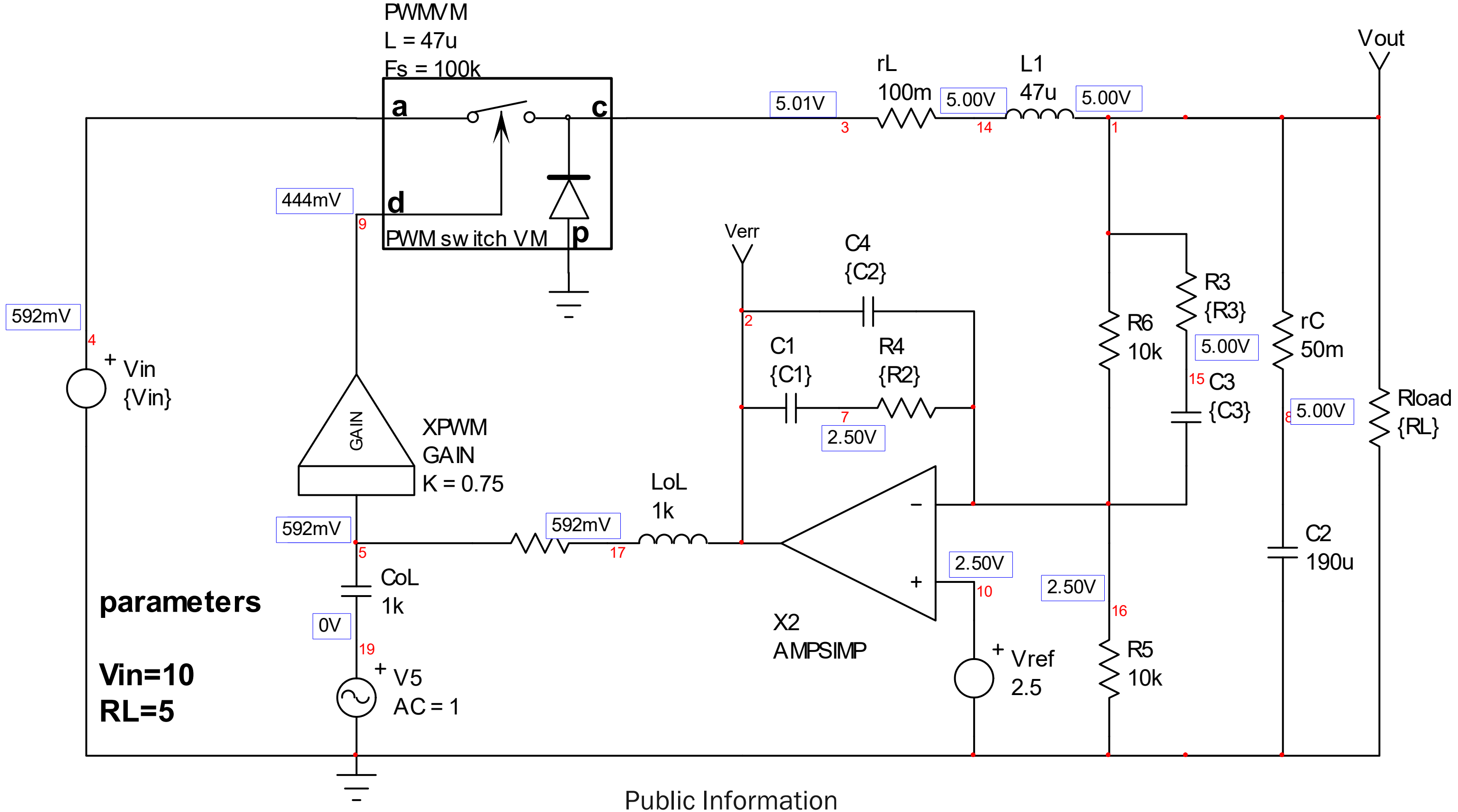
$$\omega_{z_1} = \frac{1}{R_2 C_1} \qquad \omega_{z_2} = \frac{1}{(R_1+R_3)C_3} \qquad \omega_{p_2} = \frac{1}{R_3 C_3} \qquad \omega_{p_1} = \frac{1}{R_2 \frac{C_1 C_2}{C_1+C_2}}$$

$$R_2 = \frac{R_1 f_{p_1}}{G_{f_c} f_{p_1} - f_{z_1}} \frac{\sqrt{1+\left(\frac{f_c}{f_{p_1}}\right)^2}\sqrt{1+\left(\frac{f_c}{f_{p_2}}\right)^2}}{\sqrt{1+\left(\frac{f_{z_1}}{f_c}\right)^2}\sqrt{1+\left(\frac{f_c}{f_{z_2}}\right)^2}}$$

$$C_2 = \frac{C_1}{2\pi f_{p_1} C_1 R_2 - 1} \qquad C_1 = \frac{1}{2\pi f_{z_1} R_2} \qquad C_3 = \frac{f_{p_2} - f_{z_2}}{2\pi R_{upper} f_{p_2} f_{z_2}} \qquad R_3 = \frac{R_1 f_{z_2}}{f_{p_2} - f_{z_2}}$$

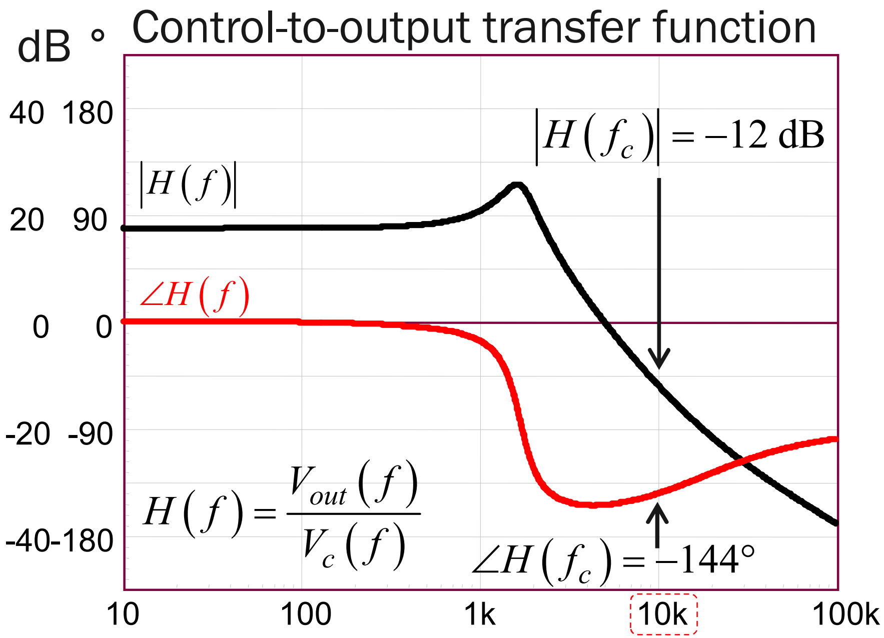
Design Example with a Type 3 Compensator

- A voltage-mode-controlled buck converter is a typical usage for a type 3



Extract Data from the Control-to-Output Transfer Function

- Choose f_c beyond f_0 but watch for the op-amp gain-bandwidth product



parameters

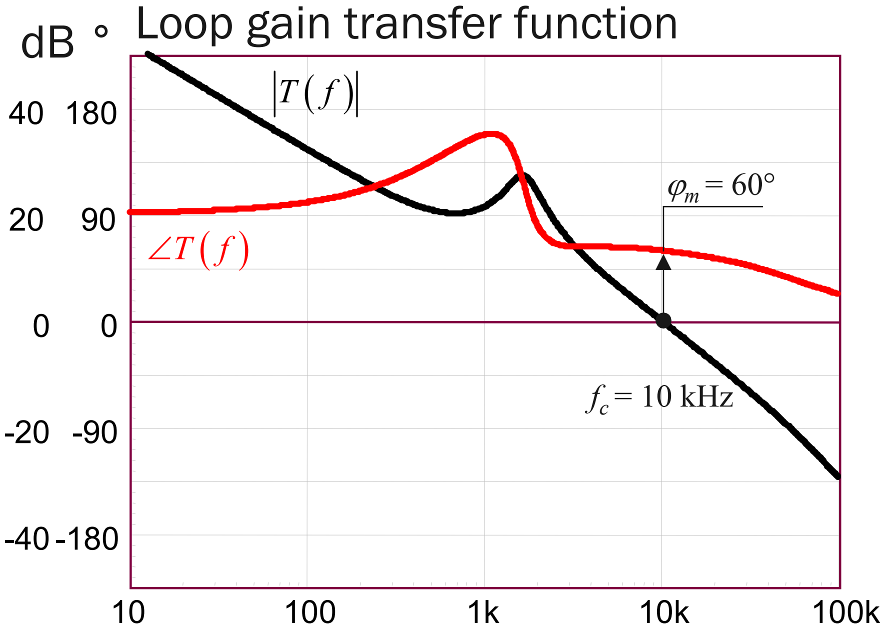
Rupper=10k
fc=10k
Gfc=-12
pfc=-144
pm=60
boost=pm-(pfc)-90
G=10^(-Gfc/20)
pi=3.14159
fz1=1k
fz2=1k
fp1=fc/tan((2*atan(fc/fz1)-tan(fc/fp2))-boost*pi/180)
fp2=50k
C1=1/(2*pi*fz1*R2)
C2=C1/(C1*R2*2*pi*fp1-1)
C3=(fp2-fz2)/(2*pi*Rupper*fp2*fz2)
R3=Rupper*fz2/(fp2-fz2)
a=sqrt((fc^2/fp1^2)+1)
b=sqrt((fc^2/fp2^2)+1)
c=sqrt((fz1^2/fc^2)+1)
d=sqrt((fc^2/fz2^2)+1)
R2=((a*b/(c*d))/(fp1-fz1))*Rupper*G*fp1

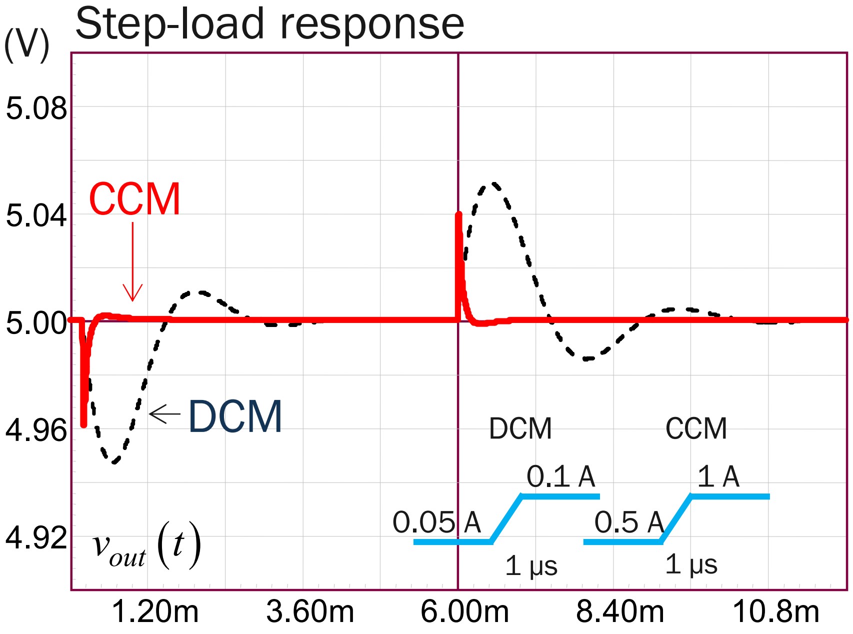
C1 = 2.63e-008
C2 = 2.70e-009
C3 = 1.56e-008
R3 = 2.04e+002
A = 1.37e+000
B = 1.02e+000
C = 1.00e+000
D = 1.00e+001
R2 = 6.06e+003

Poles and zeroes can be placed independently unlike with k-factor

Assess Compensation Effects with Transient Steps

- ❑ The buck transitions from a 2nd- to 1st-order model when going to DCM





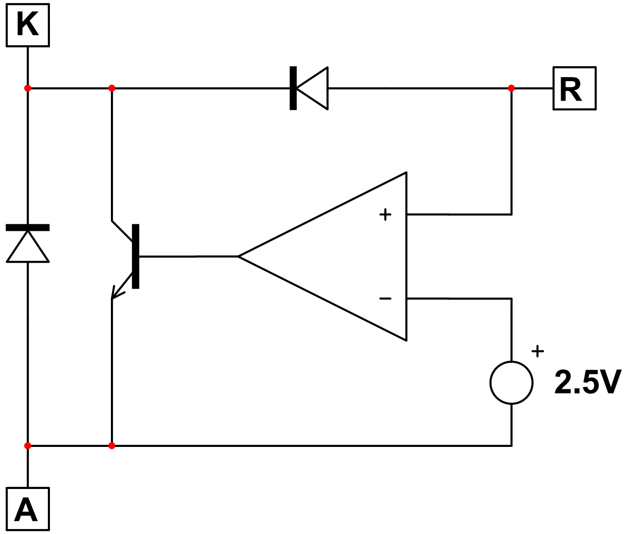
➢ One of the zeroes could go to a lower frequency and improve DCM response

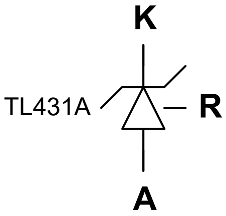
Agenda

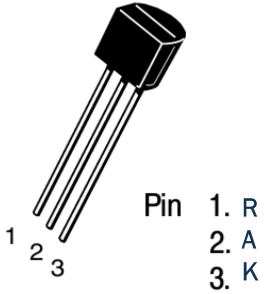
- ❑ Introduction to Control Systems
- ❑ Compensation Strategies with an Operational Amplifier
- ❑ **The TL431 at Work in Compensators**
- ❑ Building a PID
- ❑ Continuous-Time to Discrete-Time Domains
- ❑ Mapping and Compression
- ❑ Digital Compensation
- ❑ Practical Implementation of Filters
- ❑ Conclusion

The TL431 in Compensators

- The TL431 is the most popular choice in adapters designs
  - It associates an open-collector op amp and a reference voltage
  - The internal circuitry is self-supplied from the cathode current
  - When the R node exceeds 2.5 V, it sinks current from its cathode

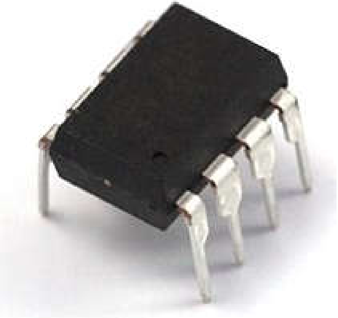


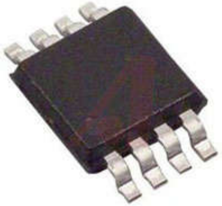


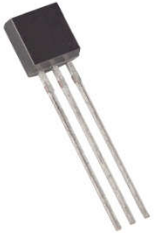


Many Different Variations Around the Part

Part number	Max voltage (V)	Min operating current (μA)	Reference voltage (V)	Packages
TL431	37	1000	2.5	TO92, SO8, μ8, DIP8
TLV431	18	100	1.24	TO92, TSOP5, SOT-23
NCP431	37	60	2.5	TO92, SO8, SOT-23
NCP100	7	100	0.9	TO92, TSOP5

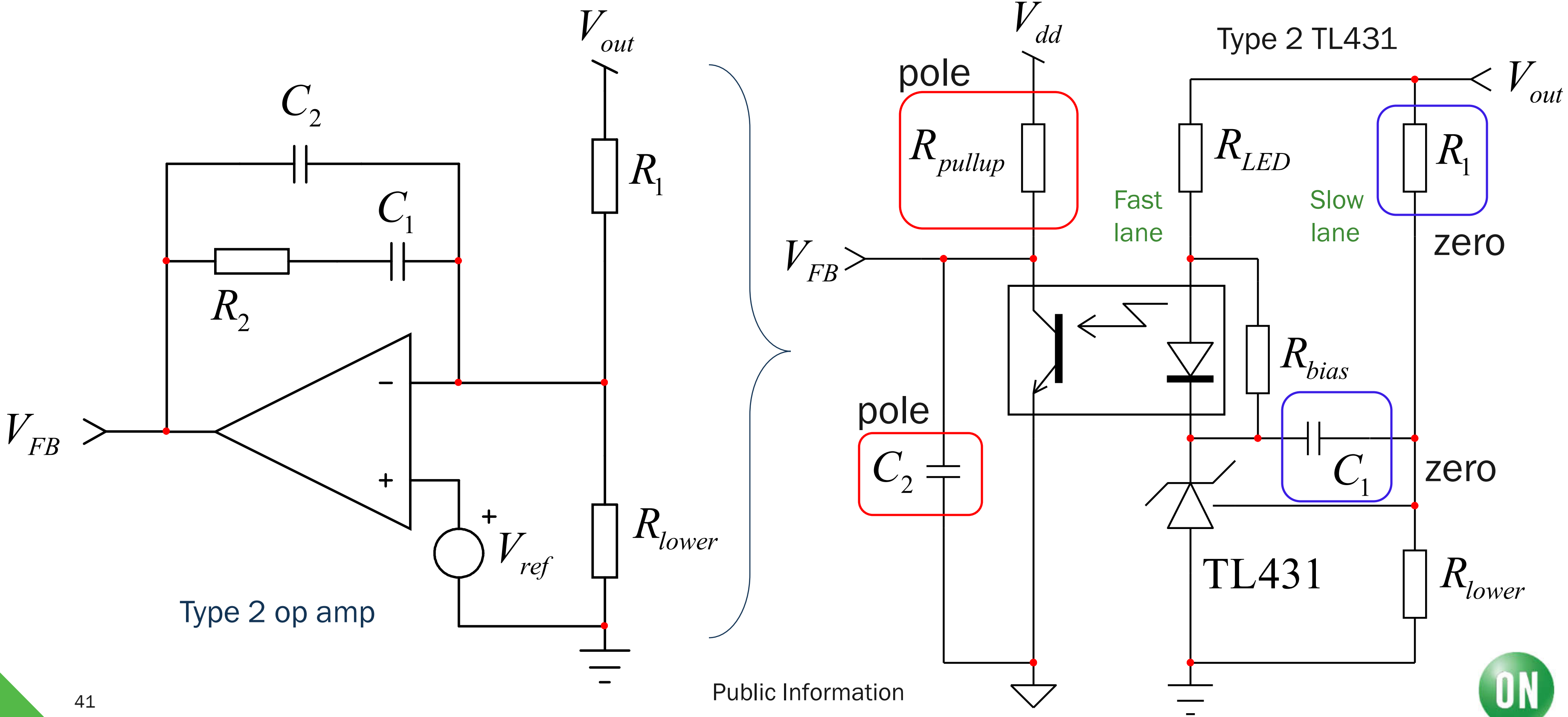






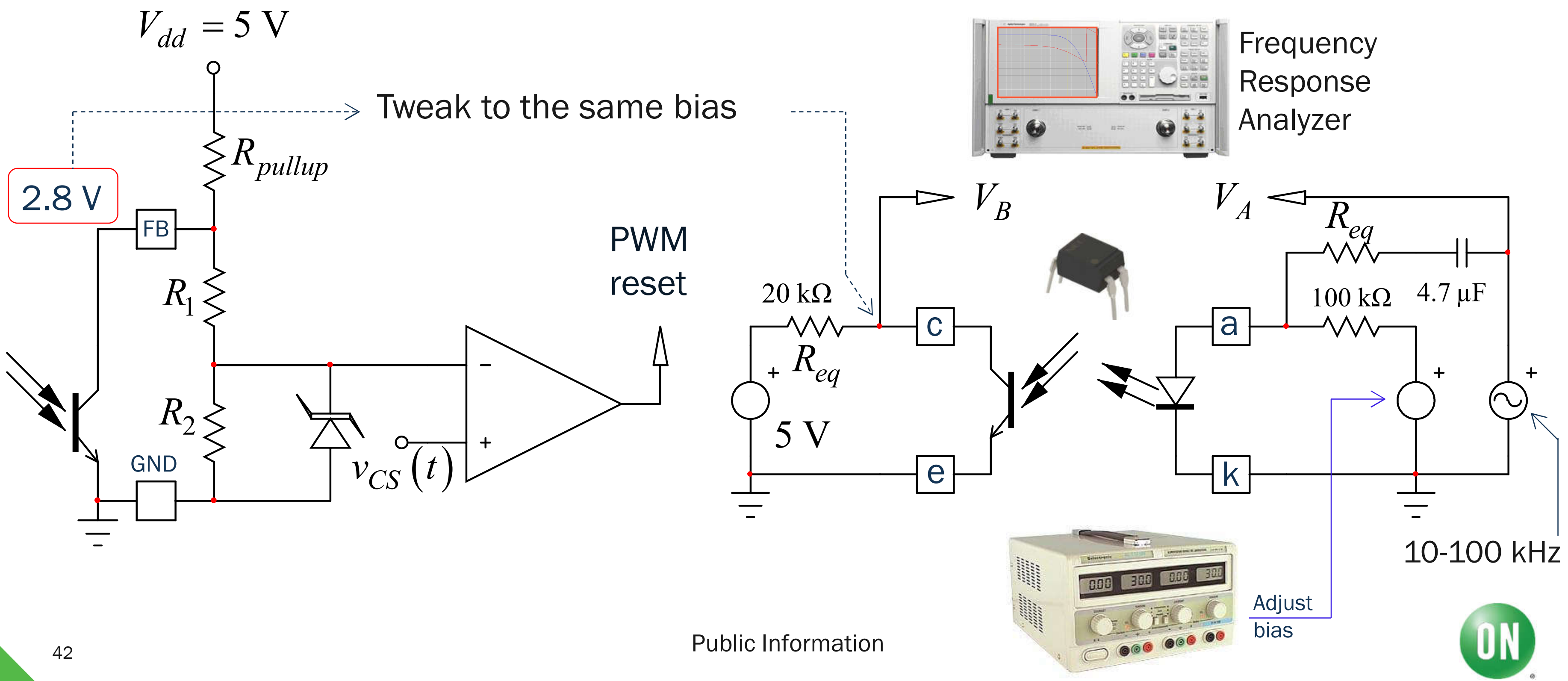
The TL431 Type 2 is Associated with the Optocoupler

- A type 2 based on the TL431 requires a single capacitor C_1



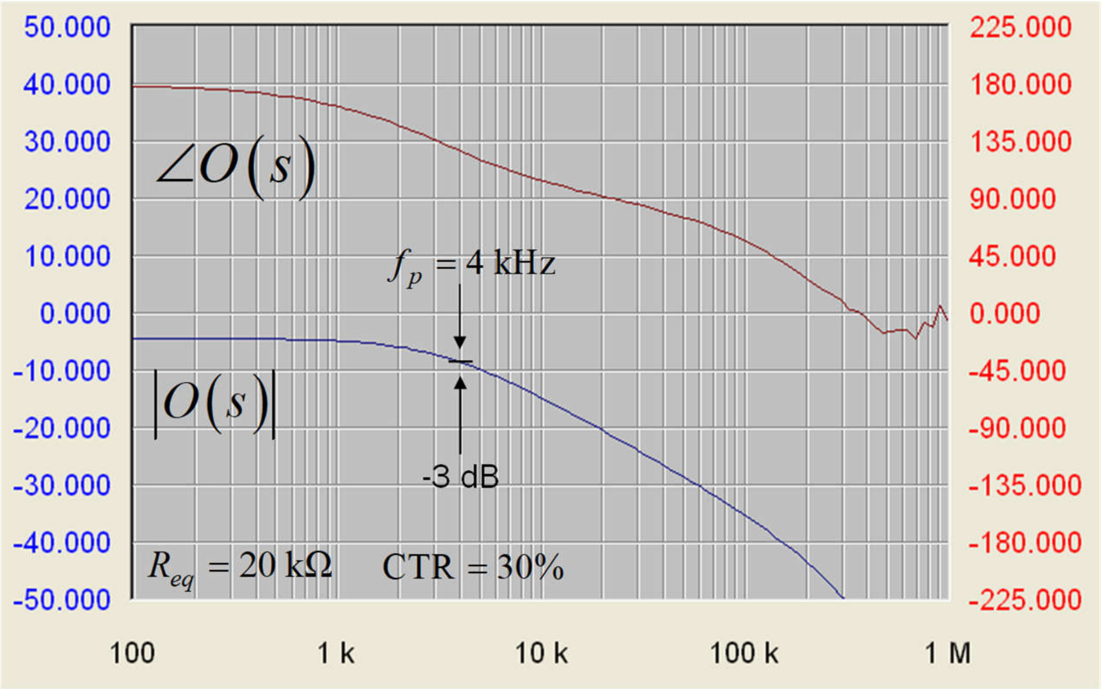
The Optocoupler Adds its Own Dynamics

- ❑ The optocoupler hosts a low-frequency pole: characterize it
- ❖ Reproduce the controller internals where the opto connects

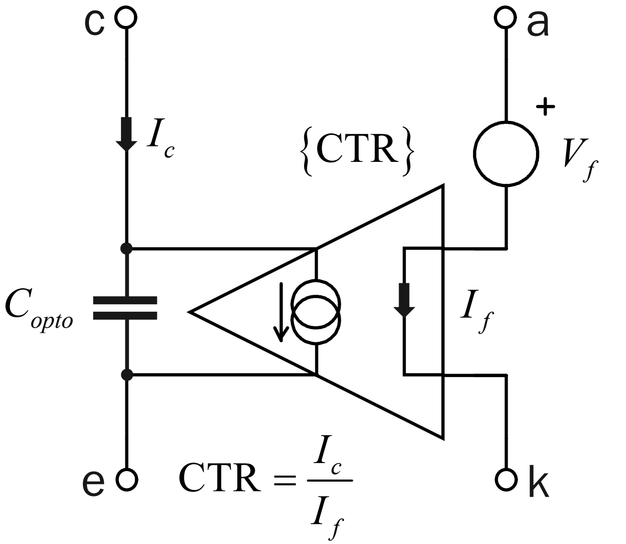


The Original Pole is Modified by the Optocoupler

- A simple 1st-order model includes an output capacitor



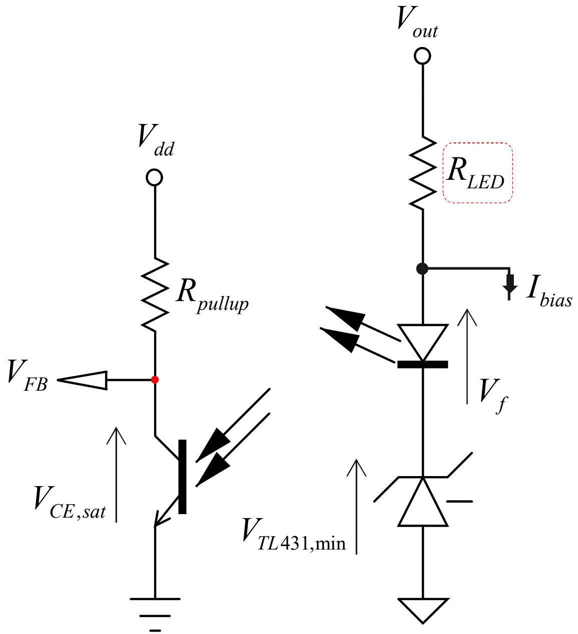
$$C_{opto} = \frac{1}{2\pi f_p R_{pullup}} \approx 2 \text{ nF}$$



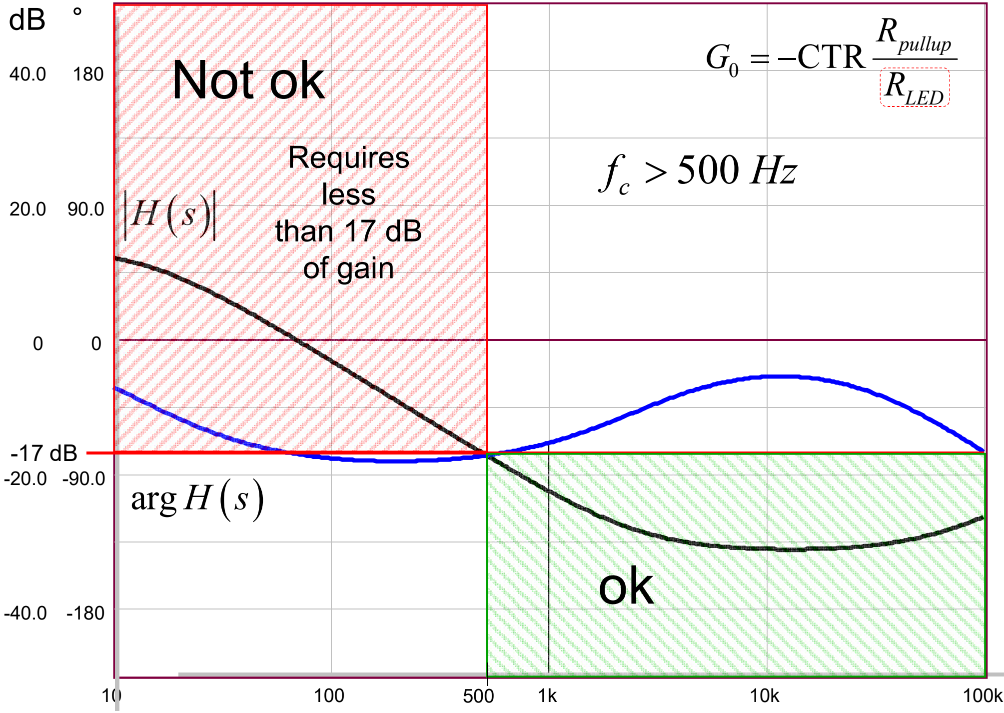
Simplified 1st-order model

The LED Resistance Influences Bias Point and Gain

- Select R_{LED} so that the optocoupler always pulls V_{FB} low in worst-case



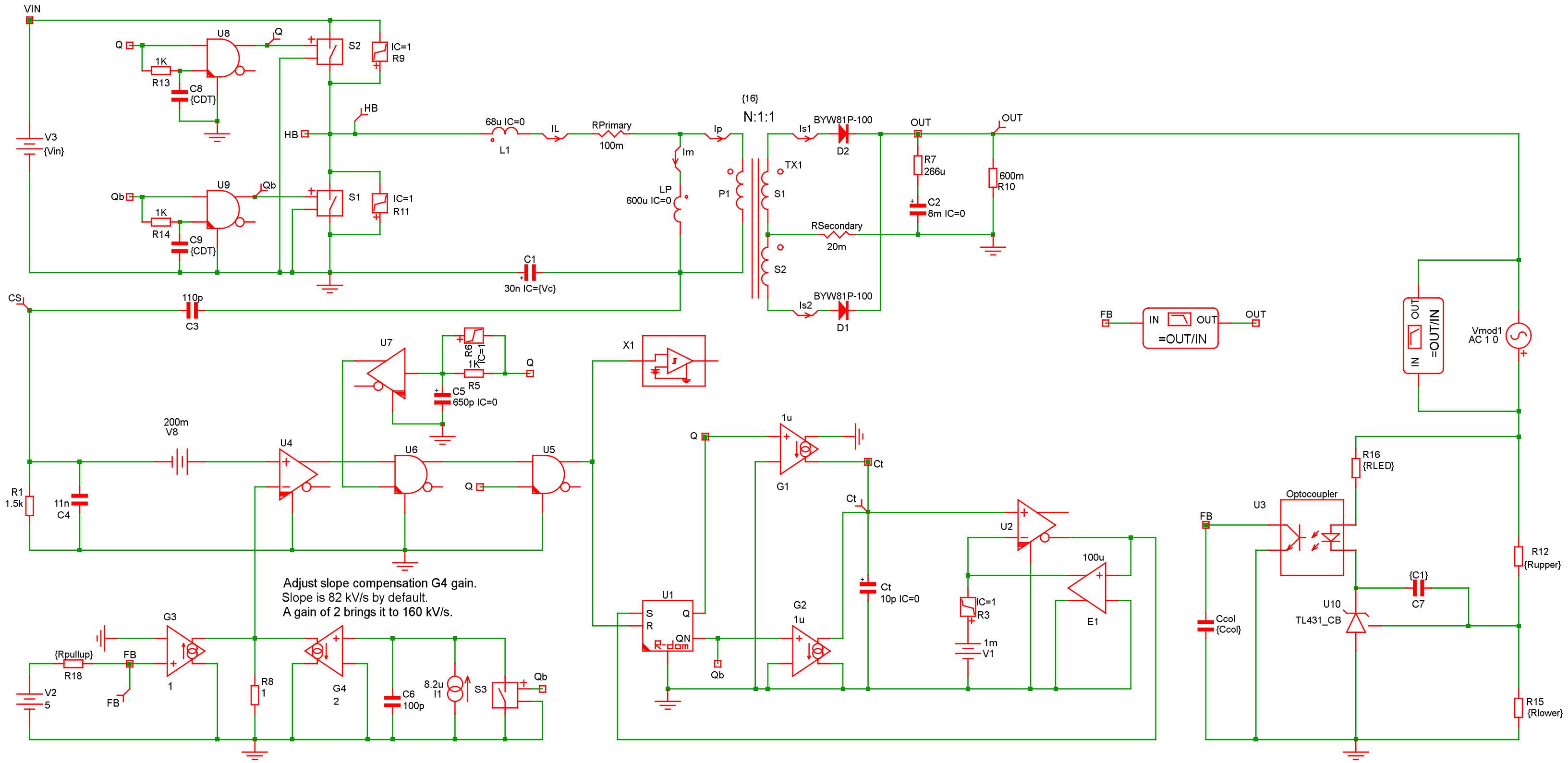
Minimum gain!



$$R_{LED,\max} \le \frac{V_{out} - V_f - V_{TL431,\min}}{V_{dd} - V_{CE,sat} + I_{bias}\mathrm{CTR}_{\min} R_{pullup}} R_{pullup}\mathrm{CTR}_{\min}$$

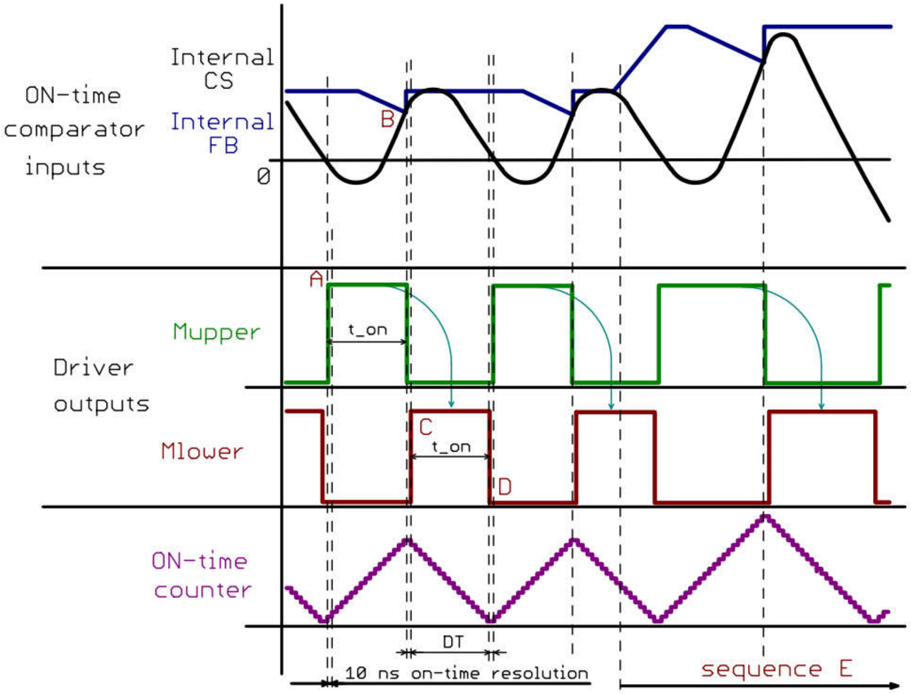
Apply the Compensation to a CM-LLC Converter

- The current mode LLC can be stabilized with a type 2 compensator

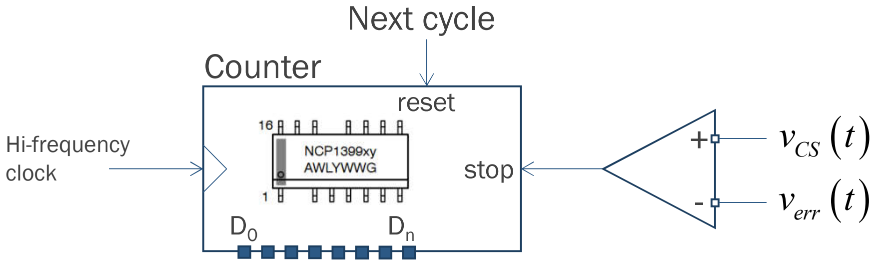


A Current-Mode LLC Converter Operates in Free-Running

- ❑ The on-time duration is precisely memorized and mirrored for the off-time



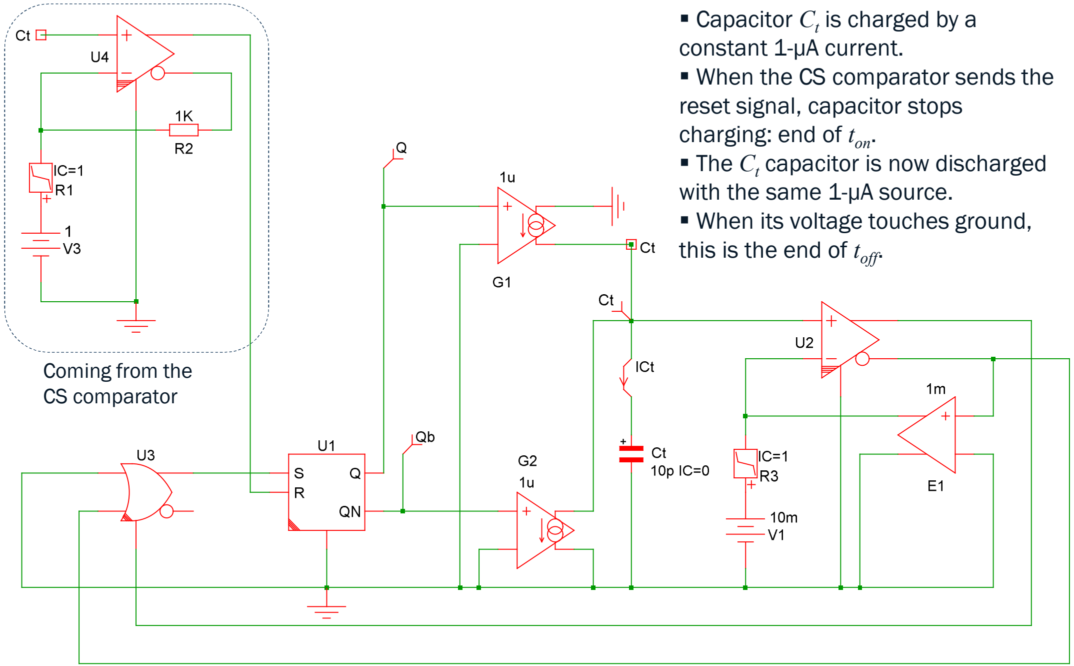
Exact 50% duty ratio



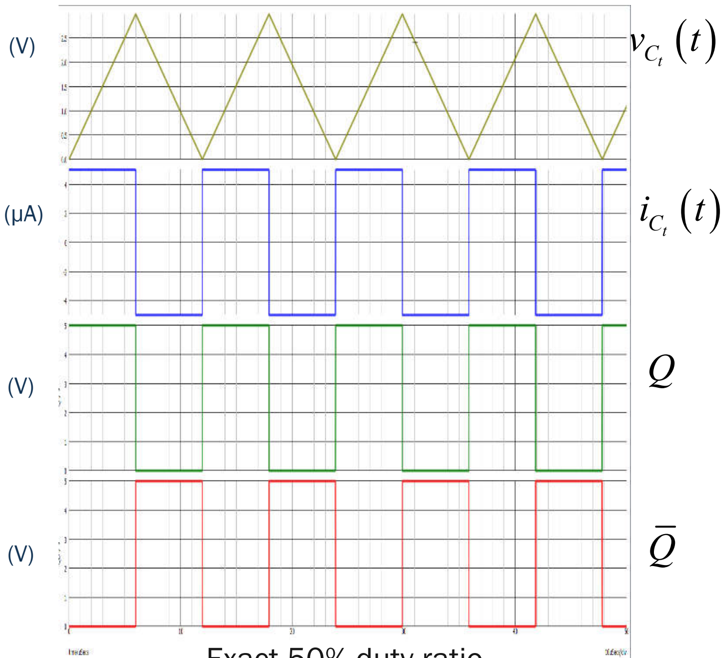
Digitized t_{on} duration

Simplified t_{on} Replication with SIMPLIS®

- ❑ Using digital counters with SIMPLIS® is not an option if small-signal analysis is wanted
  - ➢ A simple capacitor-based circuit does the job well and remains compatible with POP

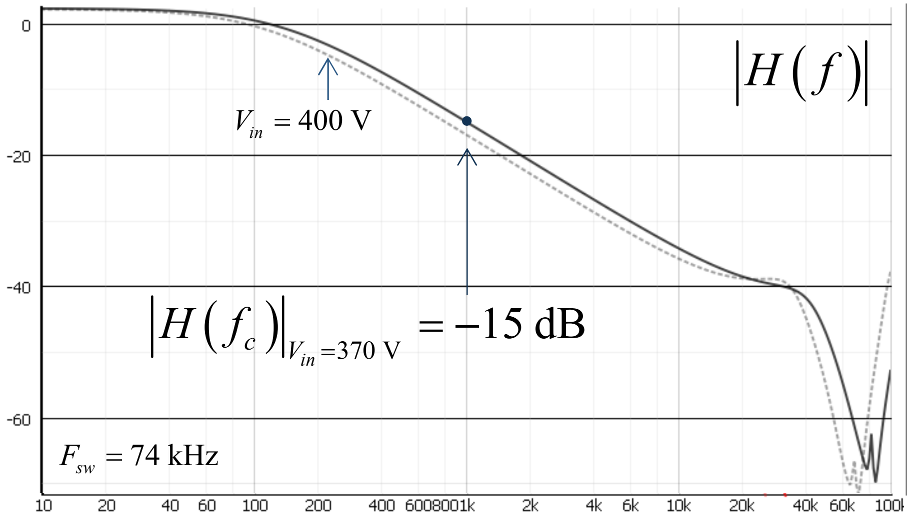


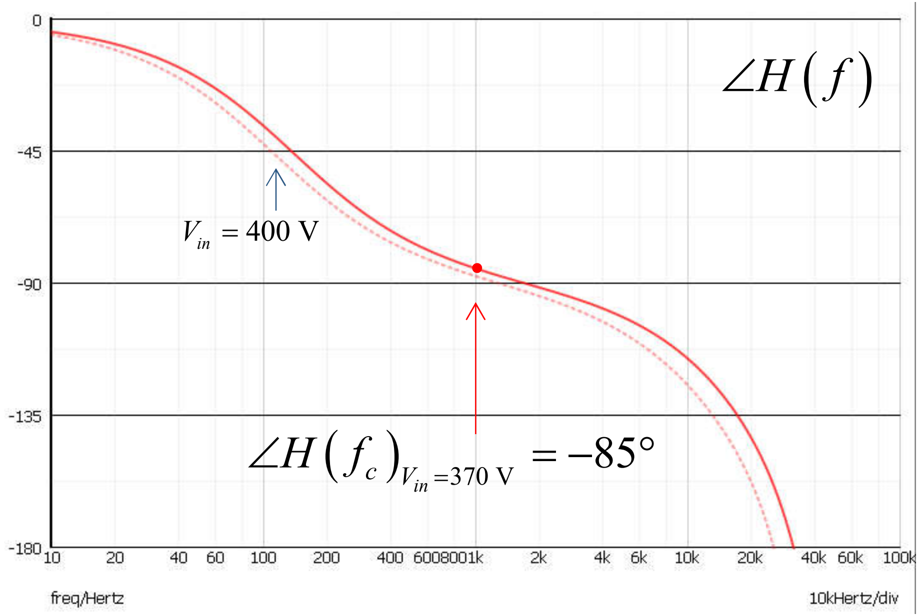
- Capacitor C_t is charged by a constant 1-µA current.
- When the CS comparator sends the reset signal, capacitor stops charging: end of t_{on}.
- The C_t capacitor is now discharged with the same 1-µA source.
- When its voltage touches ground, this is the end of t_{off}.



Exact 50% duty ratio

No Change in Plant Response at Different Inputs





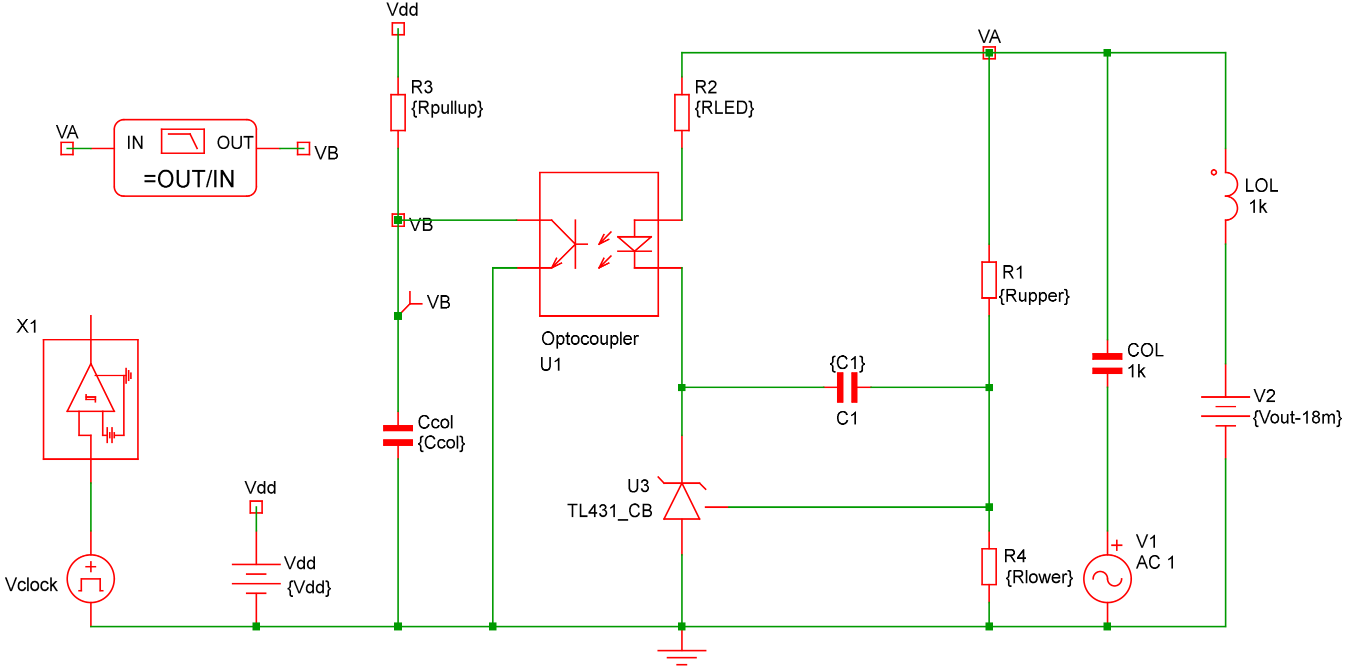
- One benefit of the CM operation is the 1st-order response.

- The control-to-output transfer function remains the same from 370 V dc to 400 V dc.

➡ For a 1-kHz crossover frequency, a simple type-2 compensator will do

Type 2 Compensator with Optocoupler

❑ Component values are calculated by SIMPLIS® to meet gain and phase boost targets

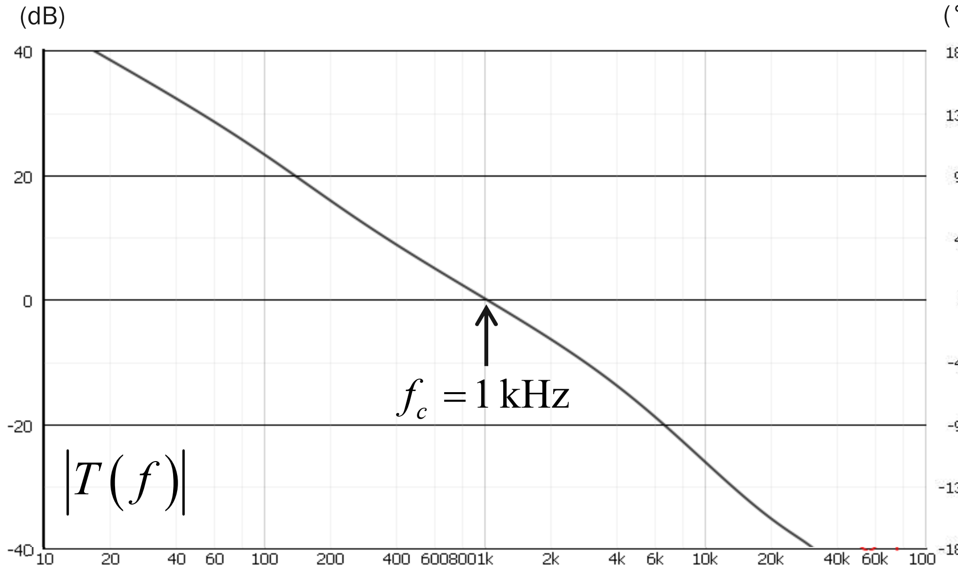


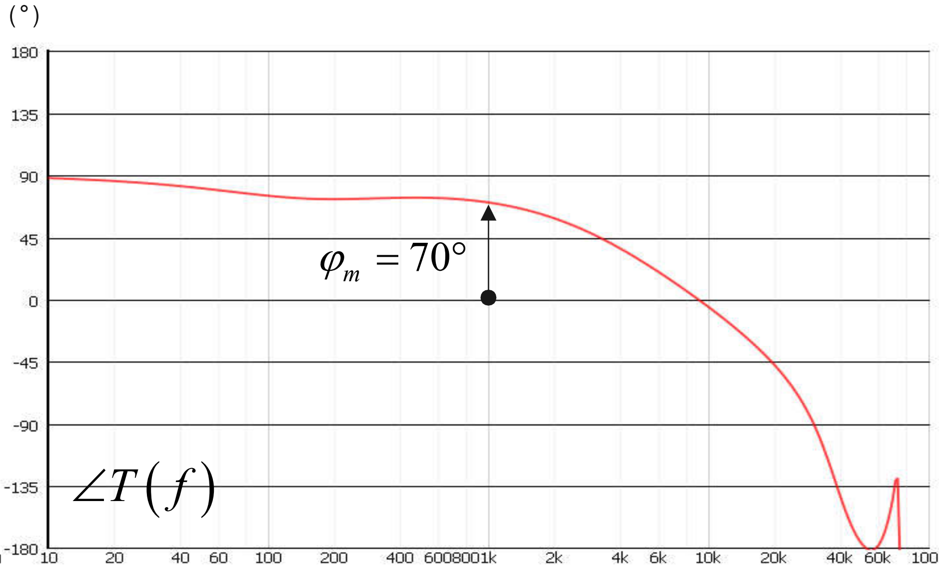

```
.GLOBALVAR Gfc=-24 * magnitude at crossover *
.GLOBALVAR PFC=-87 *plant phase at crossover *
.GLOBALVAR PM=70 * phase margin goal *
.GLOBALVAR boost=pm-(pfc)-90 * required boost *
*
*GLOBALVAR fc=1k * targeted crossover *
*
* Pole-zero calculations *
*
.GLOBALVAR k=tan((boost/2+45)*pi/180)
.GLOBALVAR fp=k*fc
.GLOBALVAR fz=fc/k
.GLOBALVAR Vout=12
.GLOBALVAR Ibias=250u
.GLOBALVAR Vref1=2.5
.GLOBALVAR Rlower=Vref1/Ibias
.GLOBALVAR Rupper=(Vout-Vref1)/Ibias
.GLOBALVAR Rpullup=18k
*
* Optocoupler specifications *
*
.GLOBALVAR Fopto=15k
.GLOBALVAR Copto=1/(2*pi*Fopto*Rpullup)
.GLOBALVAR CTR=0.3
*
```

➢ Automated calculations account for the optocoupler pole contribution

CM-LLC Compensated Loop Gain

- Plot the loop gain to check crossover frequency and phase margin at this point

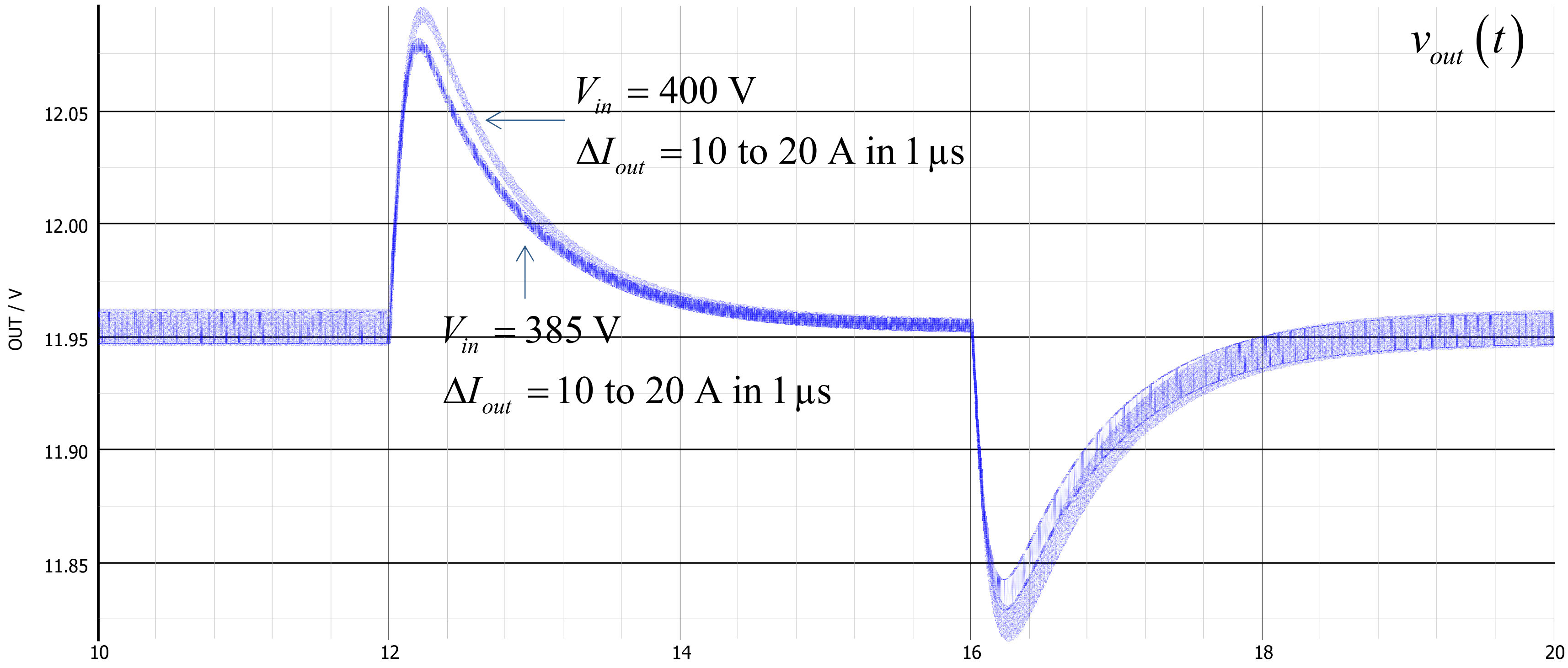




➢ You can now assess parasitics impacts by sweeping these terms in Monte Carlo runs

Compensated Transient Response

- CM makes transient response stable and immune to input voltage variations

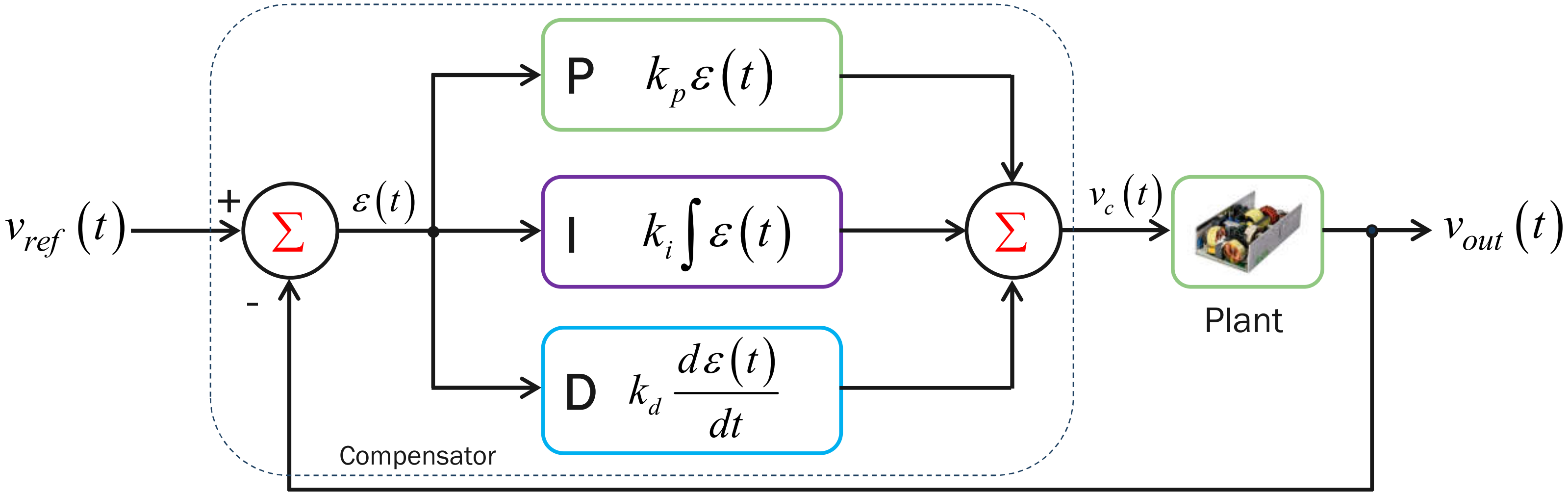


Agenda

- Introduction to Control Systems
- Compensation Strategies with an Operational Amplifier
- The TL431 at Work in Compensators
- **Building a PID**
- Continuous-Time to Discrete-Time Domains
- Mapping and Compression
- Digital Compensation
- Practical Implementation of Filters
- Conclusion

The PID Controller in the Parallel Form

- The block combines three actions: **p**roportional, **i**ntegral and **d**erivative



- Each processing block serves a purpose:

P adjusts the correction signal to the disturbance amplitude

I makes sure the static error is reduced to zero

D tailors reaction strength in relationship to disturbance speed

Many Methods to Tune a PID Compensator

- A PID can be adjusted through different approaches:

Manual

Increase ↑

Parameter	Rise time	Overshoot	Settling time	Static Error	Stability
k_p	Decrease	Increase	Small Change	Decrease	Degrade
k_i $\left[\mathrm{s}^{-1}\right]$	Decrease	Increase	Increase	Eliminate	Degrade
k_d $[\mathrm{s}]$	Minor change	Decrease	Decrease	No effect	Improve if small k_d

$$H(s)=\frac{1}{1+\frac{s}{\omega_0}+\left(\frac{s}{\omega_0}\right)^2}$$

The plant

$$G(s)=k_p+\frac{k_i}{s}+sk_d$$ Parallel form

Factor k_p

$$G(s)=k_p\left(1+\frac{1}{s\tau_i}+s\tau_d\right)$$ Standard form

The compensator

$$\tau_i=\frac{k_p}{k_i}\,[\mathrm{s}]$$

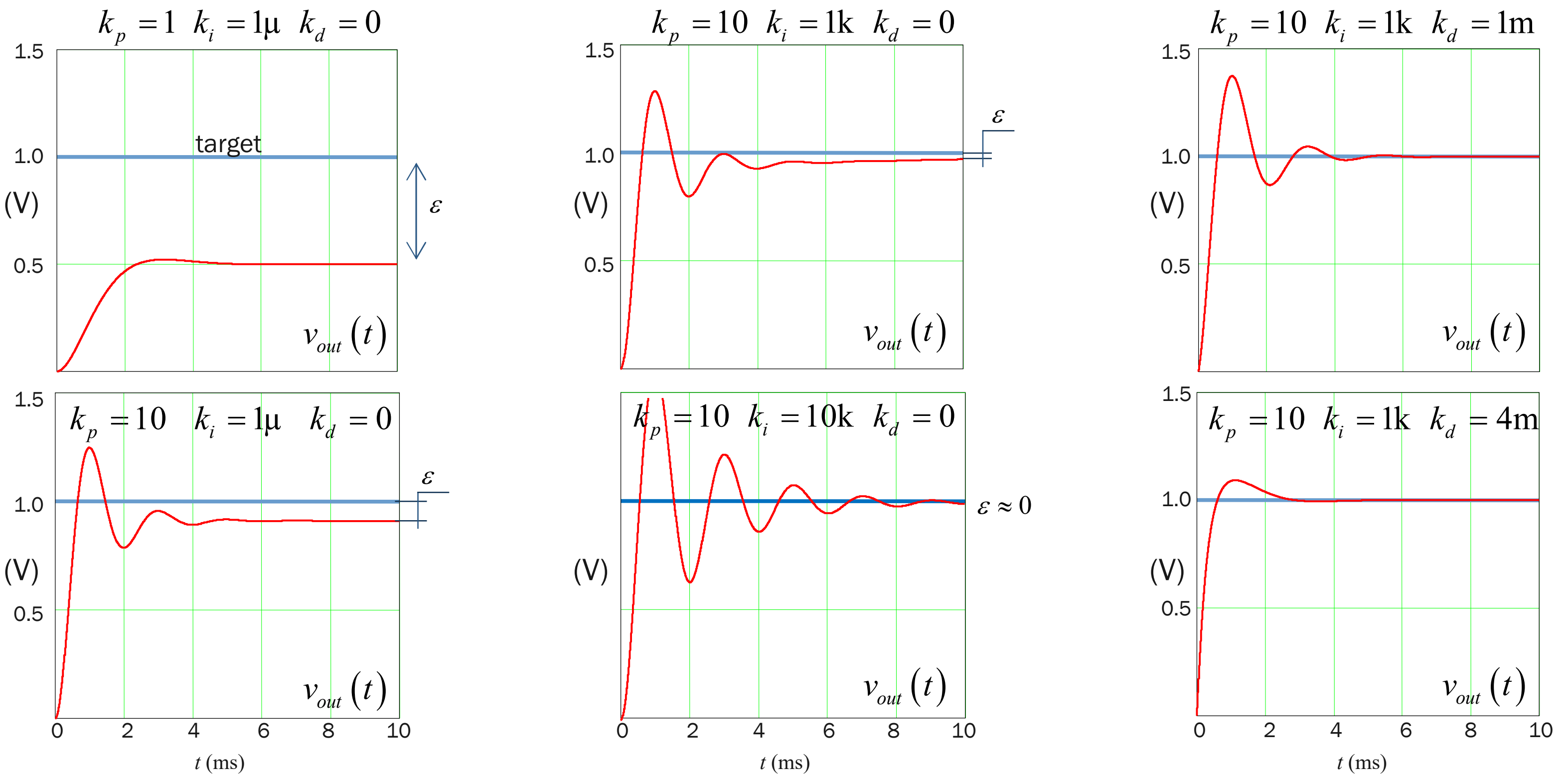
$$\tau_d=\frac{k_d}{k_p}\,[\mathrm{s}]$$

Tune k_p k_i k_d

https://en.wikipedia.org/wiki/PID_controller

Each Coefficient Affects the Transient Response

- Calculate the closed-loop response to a 1-V step input: $\dfrac{G(s)H(s)}{1+G(s)H(s)} \times \dfrac{1}{s} \rightarrow invlaplace \rightarrow v_{out}(t)$



$\omega_0 = 1\ \text{krd/s} \quad Q = 0.5$

Watch for the High-Frequency Gain

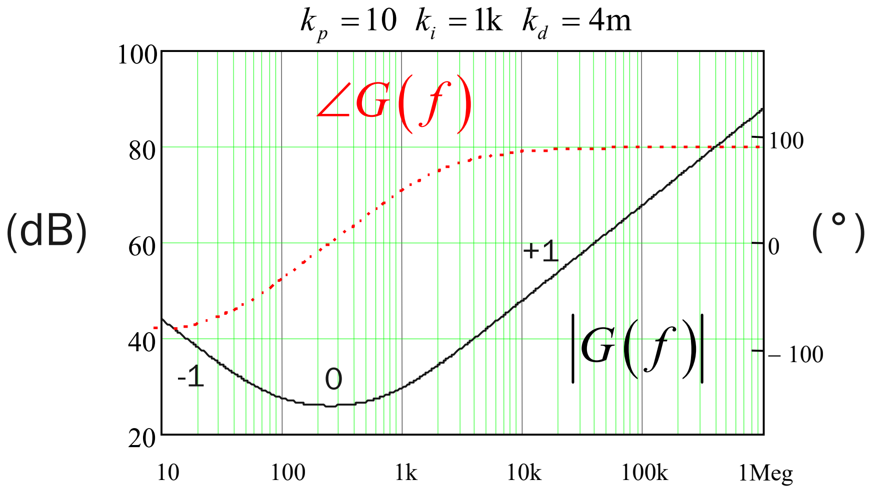
- The k_d term does not roll the gain off at high frequency and can bring noise issues

$$G(s) = k_p + \frac{k_i}{s} + sk_d$$

$$G(s) = G_0\left(1+\frac{\omega_{z_1}}{s}\right)\left(1+\frac{s}{\omega_{z_2}}\right)$$

$$\omega_{z_1} = \frac{k_p}{k_d} - \frac{k_p + \sqrt{k_p^{\,2} - 4k_d k_i}}{2k_d}$$

$$\omega_{z_2} = \frac{k_p + \sqrt{k_p^{\,2} - 4k_d k_i}}{2k_d} \qquad G_0 = \frac{k_i}{\omega_{z_1}}$$



$$\lim_{s\to\infty} s\cdot k_d \to \infty$$

The derivative term cannot be physically implemented

- Add a low-pass filter to force a flat gain at high frequency

$$G(s) = k_p\left(1+\frac{1}{s\tau_i}+s\tau_d\right) \qquad \frac{s\tau_d}{1+\frac{s\tau_d}{N}}$$

PID with pole $\omega_p = \frac{N}{\tau_d}$ → Filtered PID

Transform PID Coefficients in Poles and Zeroes

- If you add a 3rd pole, you end-up with the type 3 compensator

$$G_{\mathrm{FPID}}(s) = \left(k_p + \frac{k_i}{s} + \frac{sk_d}{1+\frac{s}{\omega_{p_1}}}\right)\frac{1}{1+\frac{s}{\omega_{p_2}}} \quad \Longleftrightarrow \quad G_{\mathrm{T3}}(s) = G_0\frac{\left(1+\frac{\omega_{z_1}}{s}\right)\left(1+\frac{s}{\omega_{z_2}}\right)}{\left(1+\frac{s}{\omega_{p_1}}\right)\left(1+\frac{s}{\omega_{p_2}}\right)}$$

$$\omega_{p_1} = \frac{1}{N\cdot k_d}$$

Filtering pole

Extra hi-frequency pole

gain

- Assume the following specifications:

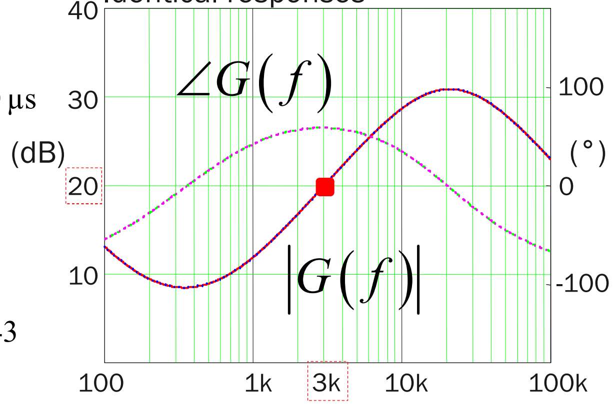
$$f_c = 3\ \mathrm{kHz}$$
$$G_{f_c} = 20\ \mathrm{dB}$$
$$f_{z_1} = 200\ \mathrm{Hz}$$
$$f_{z_2} = 600\ \mathrm{Hz}$$
$$f_{p_1} = 21\ \mathrm{kHz}$$
$$f_{p_2} = 21\ \mathrm{kHz}$$

$$G_0 = \frac{\sqrt{1+\left(\frac{f_c}{f_{p_1}}\right)^2}\sqrt{1+\left(\frac{f_c}{f_{p_2}}\right)^2}}{\sqrt{1+\left(\frac{f_{z_1}}{f_c}\right)^2}\sqrt{1+\left(\frac{f_c}{f_{z_2}}\right)^2}} \approx 2$$

$$k_d = \frac{\left(\omega_{z_2}-\omega_{p_1}\right)\left(k_i - G_0\omega_{p_1}\right)}{\omega_{p_1}{}^2\omega_{z_2}} \approx 510\ \mu\mathrm{s}$$

$$k_i = G_0\omega_{z_1} = 2.51\ \mathrm{ks}^{-1}$$

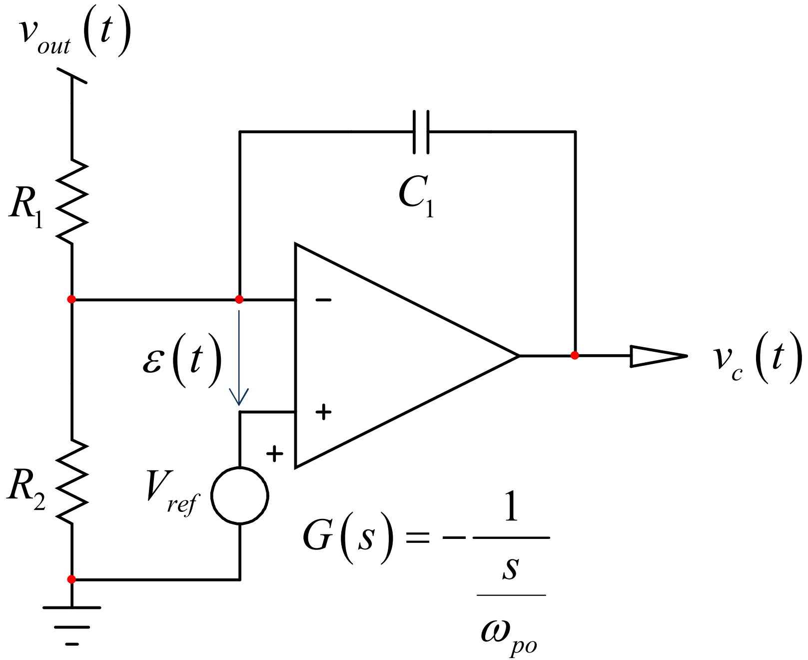
$$k_p = G_0\left(\frac{\omega_{z_1}+\omega_{z_2}}{\omega_{z_2}} - \frac{\omega_{z_1}}{\omega_{p_1}}\right) = 2.643$$



C. Basso, *Designing Control Loops for Linear and Switching Power Supplies*, Artech House, 2012

Integrator Saturation Issue

- ❑ The Integral term accumulates the error over time



$$G(s) = -\frac{1}{\frac{s}{\omega_{po}}}$$

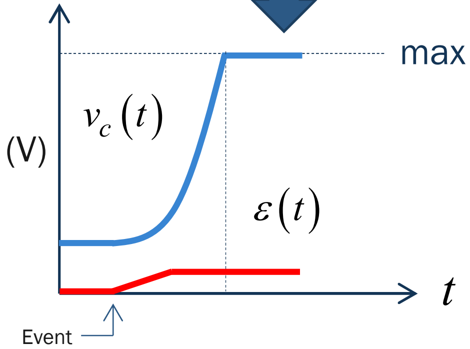
$$\omega_{po} = \frac{1}{R_1 C_1}$$

1. Assume a large output current step
2. Control asks for maximum duty ratio, e.g. 80%
3. Output cannot keep up, control voltage is flat

$$v_c(t) = \frac{k_p}{\tau_i}\int_0^t \varepsilon(\tau)\,d\tau$$

$\varepsilon(\tau)$ is constant

$$v_c(t) = \frac{k_p}{\tau_i}\varepsilon \cdot t$$



SPICE Simulates the Saturation Effect

- ❑ An average model coupled to a PID simulates a voltage-mode buck converter

parameters

fc=10k
Gfc=-20
Vin=10
Vpeak=2

G=10^(-Gfc/20)
pi=3.14159

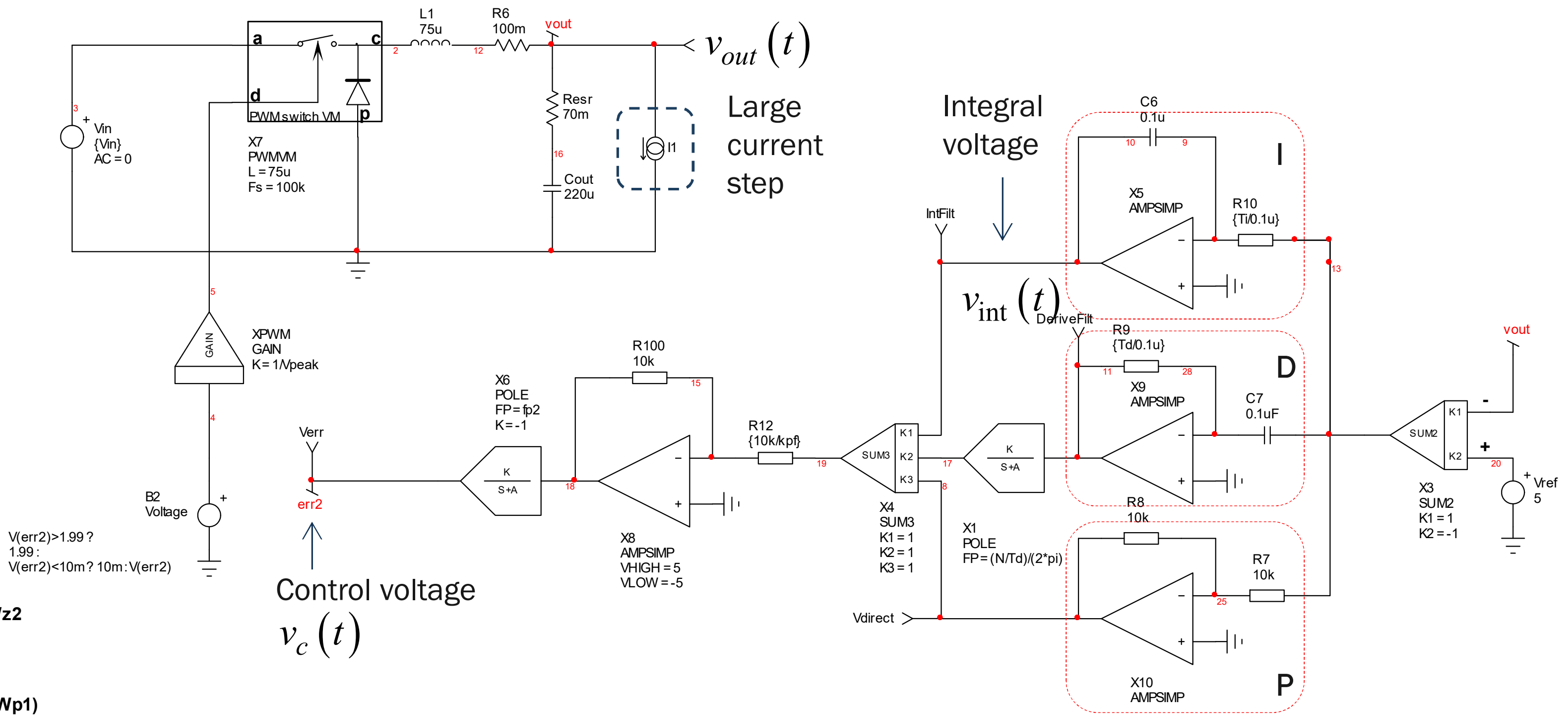
fz1=1.2k
fz2=1.2k
fp1=11.3k
fp2=50k

Wz1=2*pi*fz1
Wz2=2*pi*fz2
Wp1=2*pi*fp1

i=(1+fc^2/fp1^2)*(1+fc^2/fp2^2)
j=(1+fz1^2/fc^2)*(1+fc^2/fz2^2)
Wpi=sqrt(i/j)*G*fz1*2*pi

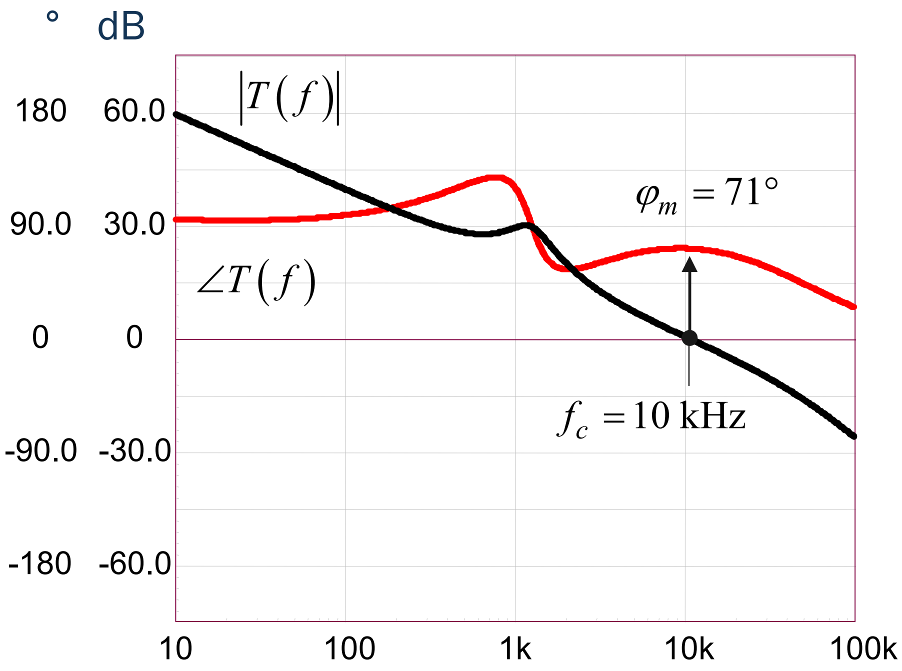
e=(Wp1-Wz1)*(Wp1-Wz2)
f=Wp1*Wz1+Wp1*Wz2-Wz1*Wz2

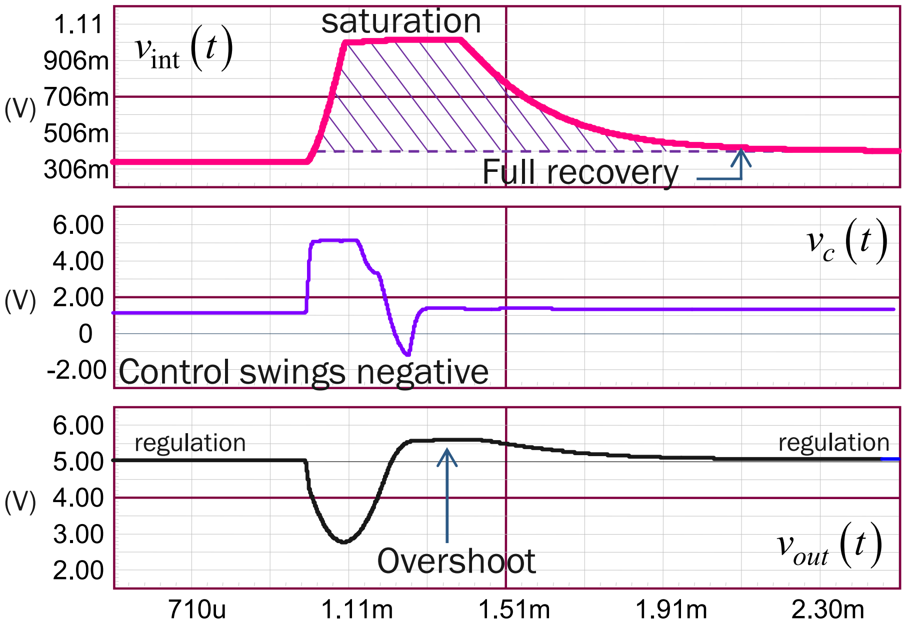
Td=e/(f*Wp1)
N=((Wp1^2)/f)-1
Ti=((Wz1+Wz2)/(Wz1*Wz2))-(1/Wp1)
kpf=(Wpi/Wz1)-(Wpi/Wp1)+(Wpi/Wz2)



Windup Phenomenon in the Integral Path

❑ The small-signal response shows a 10-kHz bandwidth and a good phase margin

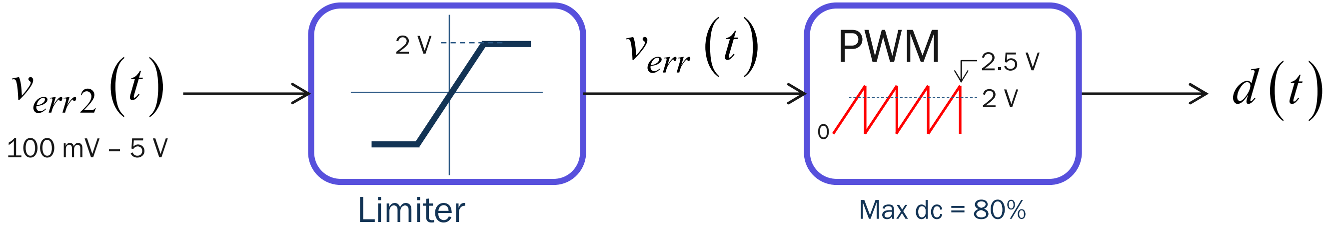




❖ The transient response shows overshoot due to saturation and recovery time

Anti-Windup Methods

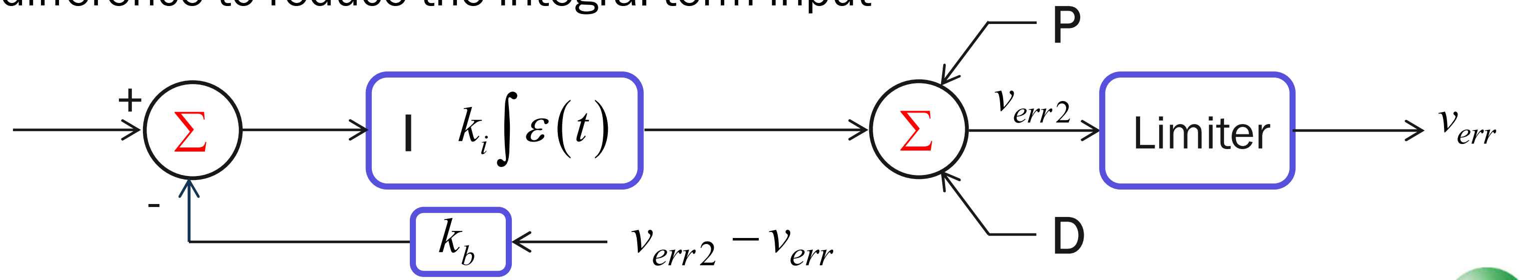
- ❑ Insert a clamping circuit at the PID output



Normal operation: $v_{err2}(t) - v_{err}(t) = 0$

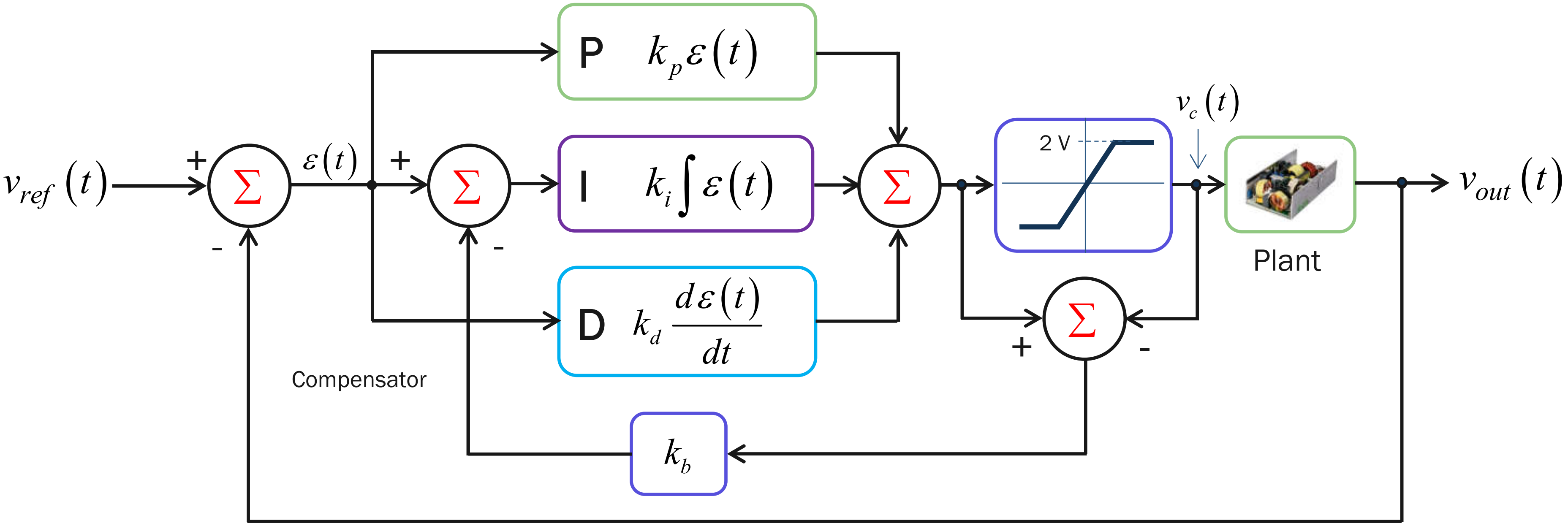
Clamp active: $v_{err2}(t) - v_{err}(t) \neq 0$

- ❑ Scale the difference to reduce the integral term input



The PID with Anti-Windup Back Calculation

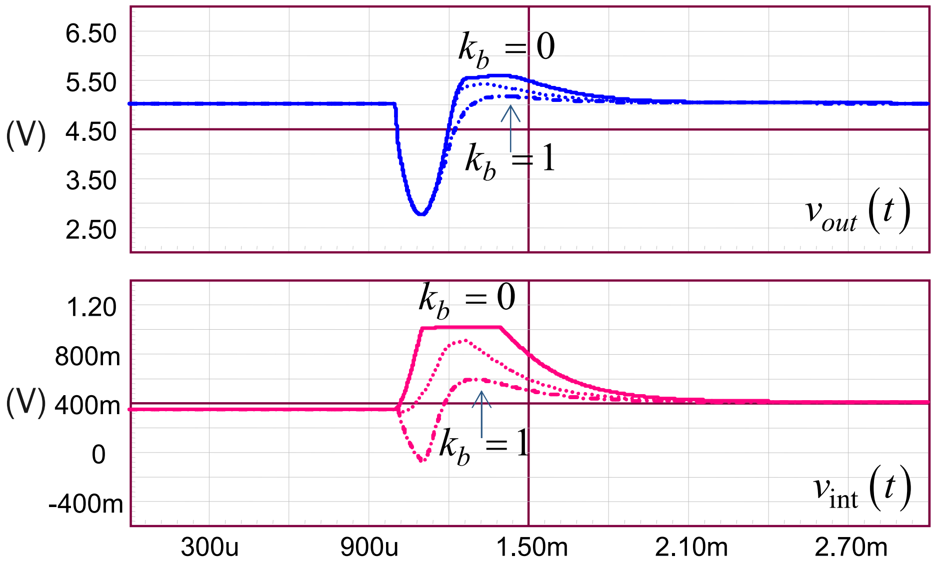
- ❑ The injection in the integral path reduces its voltage excursion

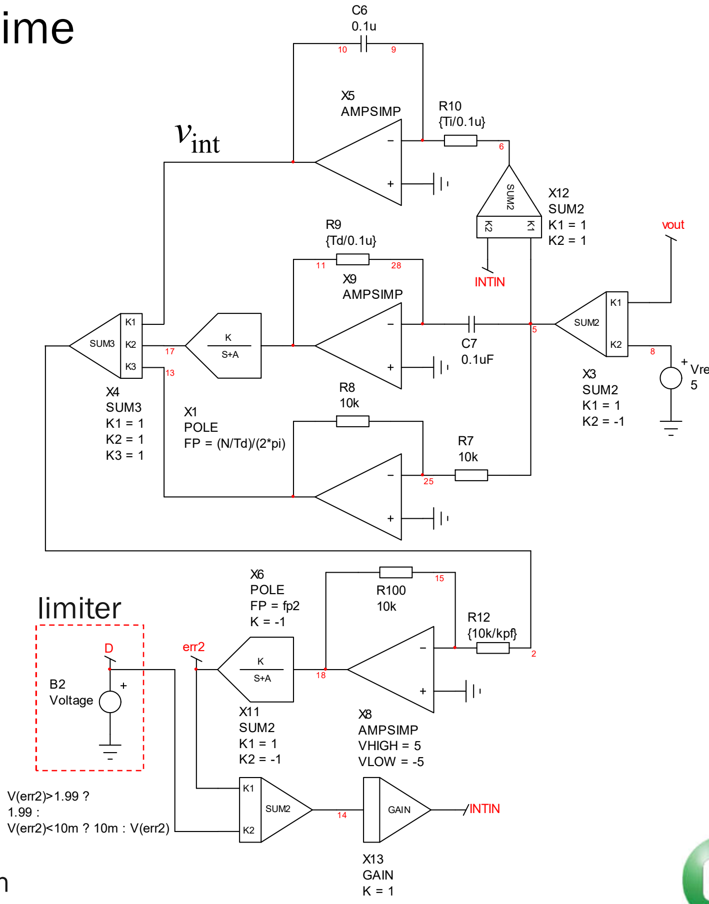


➢ Adjust the k coefficient to meet the overshoot specs

Transient Response with Back Calculation Anti-Windup

- The overshoot is gone with almost no recovery time

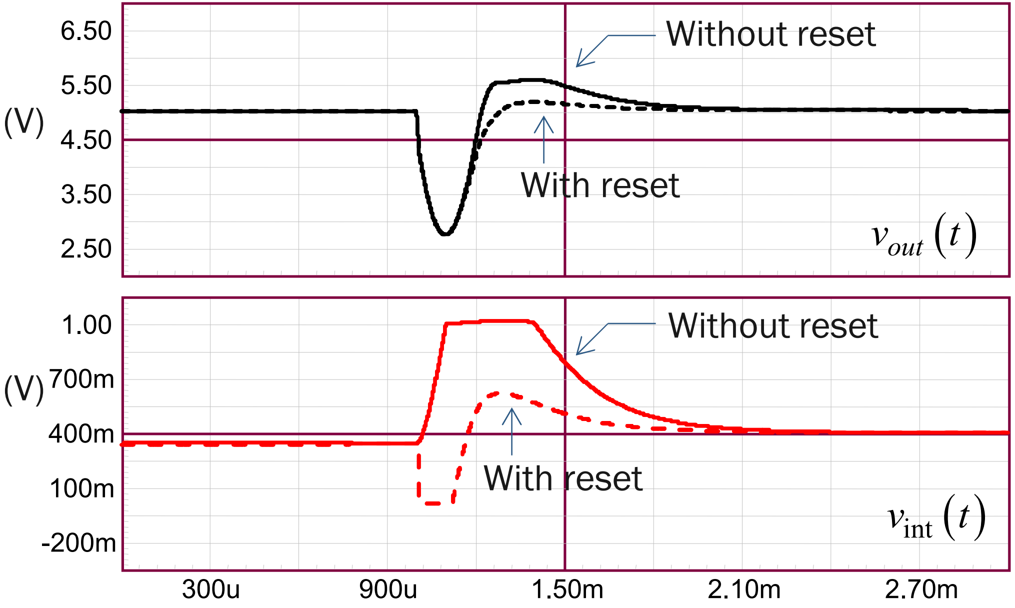


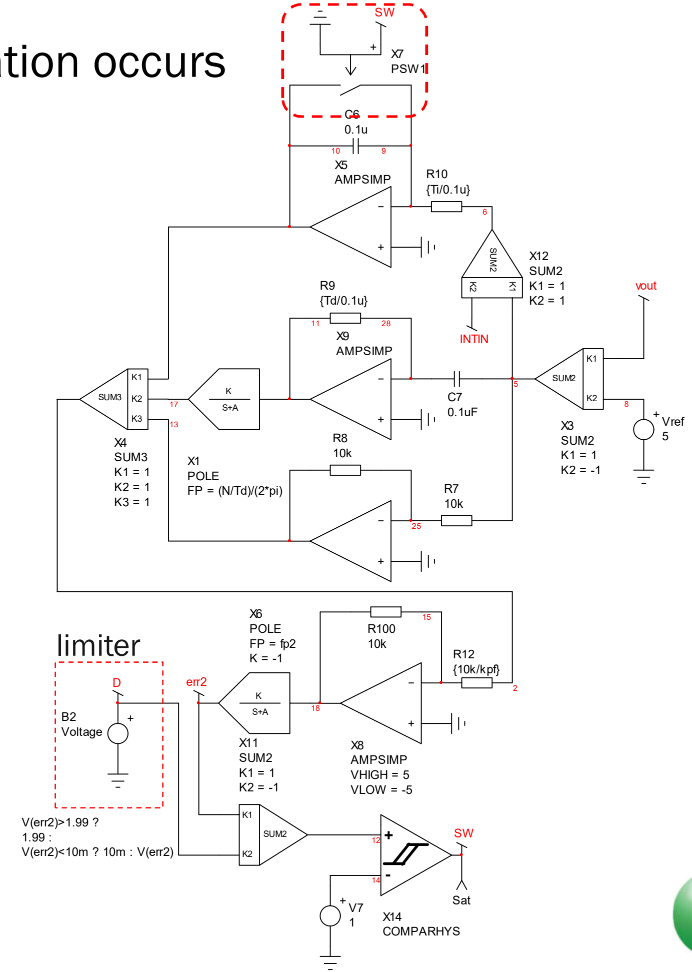


- Increasing k too much degrades response

Transient Response with Integral Reset Anti-Windup

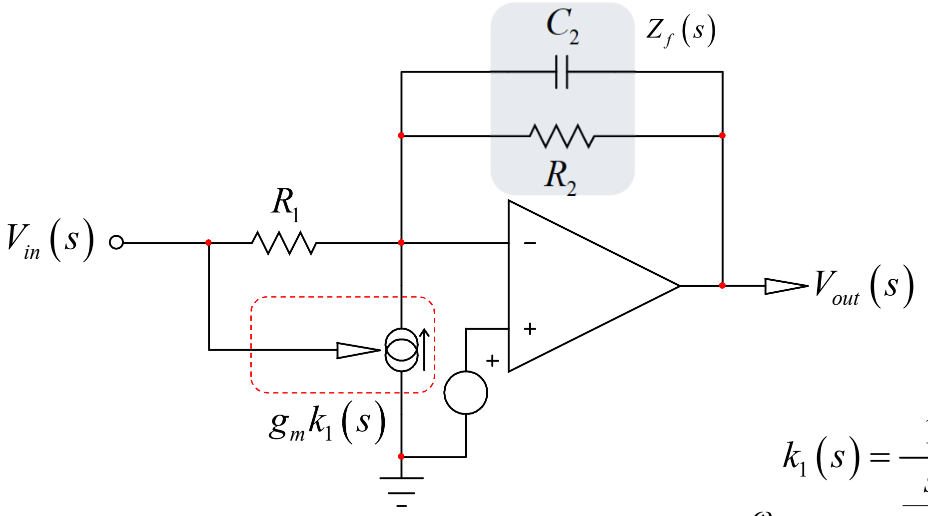
- A switch can reset the integral path when saturation occurs





Turn off the Pole at the Origin in a Type 2

- The added current source can be turned on or off, removing the pole at the origin



$$Z_f(s) = R_2 \parallel \frac{1}{sC_2} = R_2 \frac{1}{1+\frac{s}{\omega_p}} \qquad \omega_p = \frac{1}{R_2 C_2}$$

$$G(s) = -\left(g_m k_1(s) \frac{R_2}{1+\frac{s}{\omega_p}} + G_0 \frac{1}{1+\frac{s}{\omega_p}} \right) \qquad G_0 = \frac{R_2}{R_1}$$

$$k_1(s) = \frac{1}{s}$$

$$-G_0 \frac{1+\frac{R_2}{G_0} g_m k_1(s)}{1+\frac{s}{\omega_p}} = -G_0 \frac{1+\frac{\omega_z}{s}}{1+\frac{s}{\omega_p}}$$

New equation | Type 2

$$\omega_{po} = \frac{\omega_z}{g_m R_1} \longrightarrow \omega_{po}$$

- The technique can be extended to a type 3 compensator

Example with the Pole at the Origin Toggling Circuit

- The added current source can be turned on or off, removing the pole at the origin

parameters

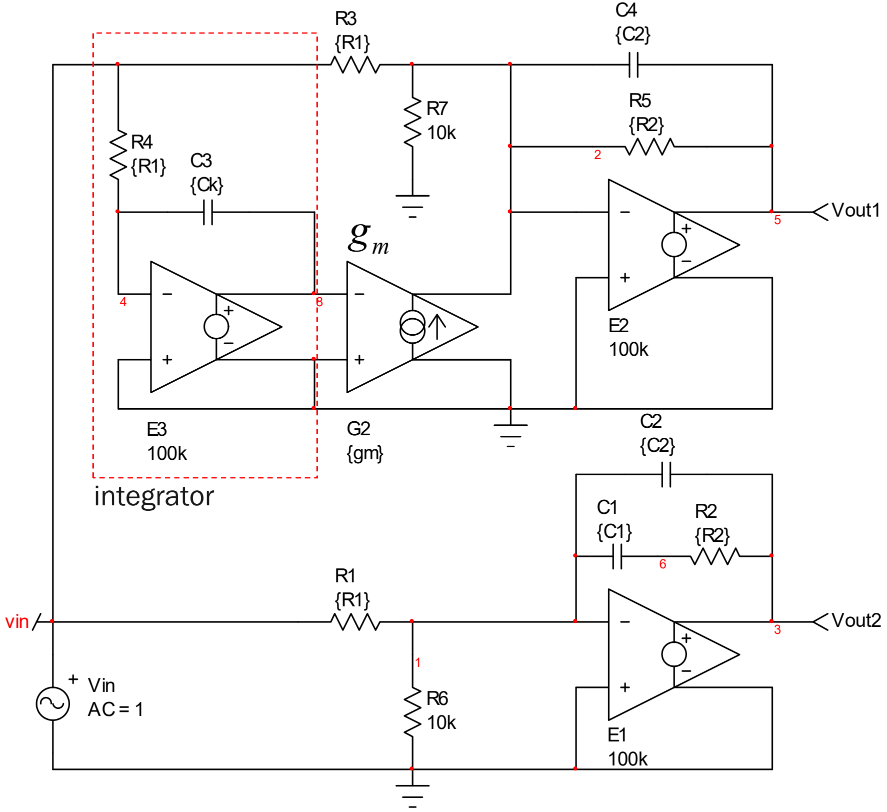
R1=38k
fc=1k
Gfc=-20
pfc=-90
pm=70

G=10^(-Gfc/20)
boost=pm-(pfc)-90
pi=3.14159
K=tan((boost/2+45)*pi/180)

fz=fc/k
fp=k*fc

gm=10u
R2=G*R1
C2=1/(2*pi*fp*R2)
C1=1/(2*pi*R2*fz)
Ck=gm/(2*pi*fz)

Approximate formulas for $C_2 << C_1$

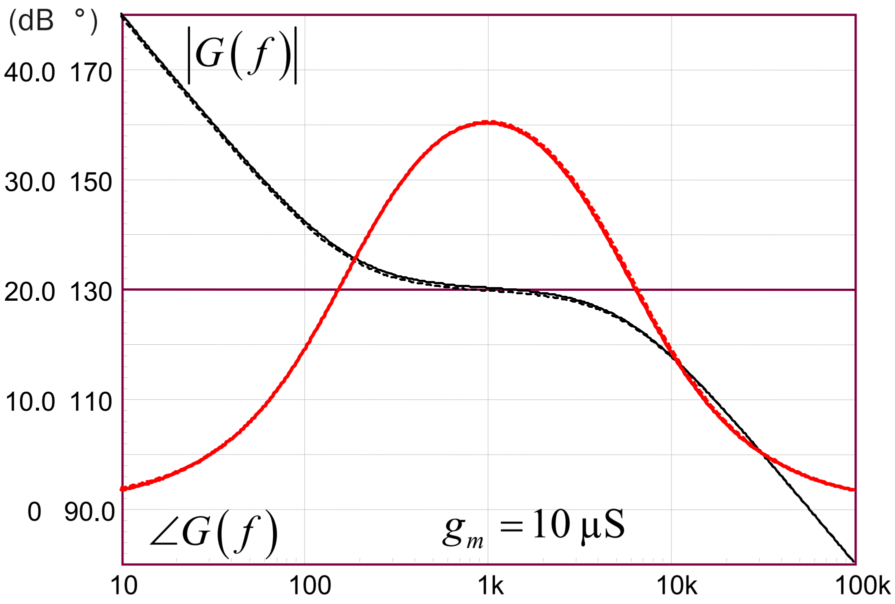


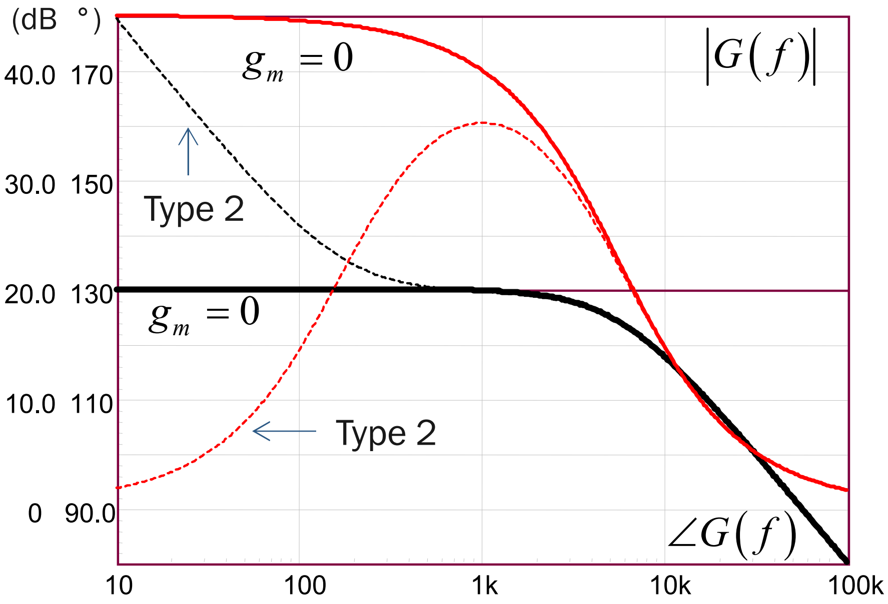
Pole/zero on demand

Classical type 2 compensator

On-Demand Pole at the Origin and Zero

- By setting the transconductance g_m to zero, the pole and zero are removed





- Turn the current source off to improve overshoot and recovery time

Application in a High-Current Dc-Dc Converter

- The output current changes from 61 A to 1 A in 1 µs: undershoot and ringing are gone

Pole at the origin is active

NCP81111

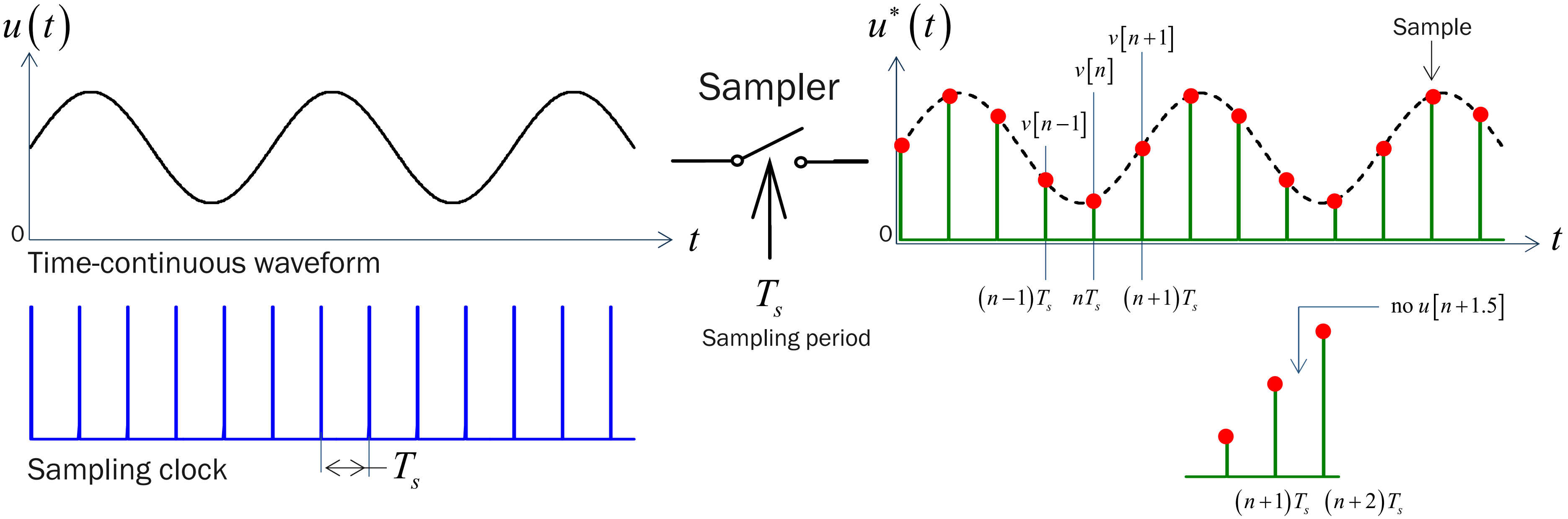
Pole at the origin is disconnected and back in place

Agenda

- ❑ Introduction to Control Systems
- ❑ Compensation Strategies with an Operational Amplifier
- ❑ The TL431 at Work in Compensators
- ❑ Building a PID
- ❑ Continuous-Time to Discrete-Time Domains
- ❑ Mapping and Compression
- ❑ Digital Compensation
- ❑ Practical Implementation of Filters
- ❑ Conclusion

Continuous-Time and Discrete-Time Signals

- The switch samples the input signal at a T_s switching interval

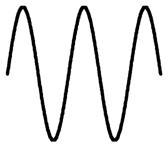


- There is no discontinuity in u along the time axis
  - u can take on any value: $u(t) = U \cdot \sin(\omega t)$
- There is no in-between sample value
  - n is an integer: $u^*(n) = U \cdot \sin(nT_{sw})$

Fourier, Laplace and z-Transforms

- Laplace and Fourier transforms map continuous-time functions

Time-continuous



$u(t)$

Fourier transform

$$U(\omega)=\int_{-\infty}^{\infty} u(t)\cdot e^{-j\omega t}\,dt$$

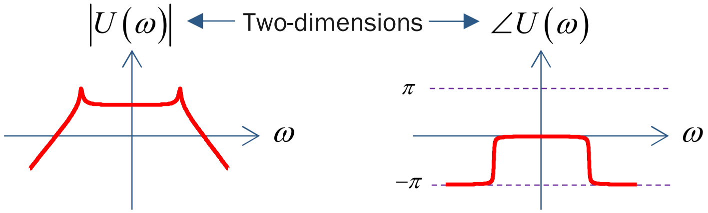
bilateral

Laplace transform

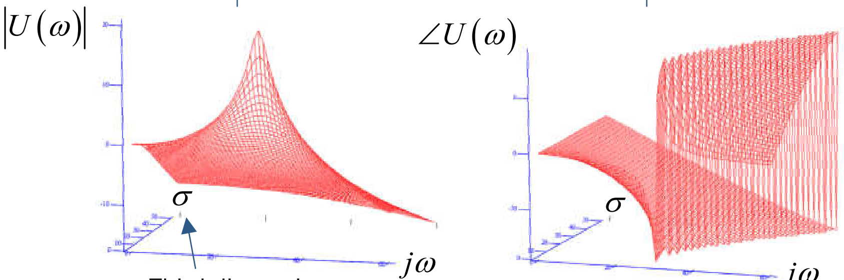
$$U(s)=\int_{0}^{\infty} u(t)\cdot e^{-st}\,dt \qquad s=\sigma+j\omega$$

unilateral

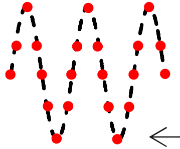
frequency domain



s-domain



- The z-transform maps discrete-time functions



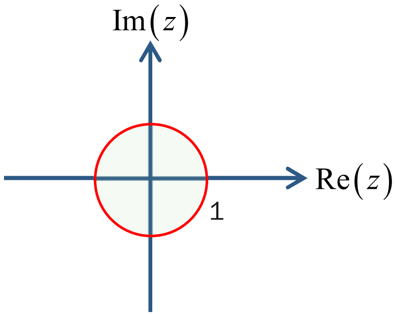
$u^*(t)$

$$U(z)=\sum_{n=0}^{\infty} u[n]z^{-n}$$

Discrete sample value

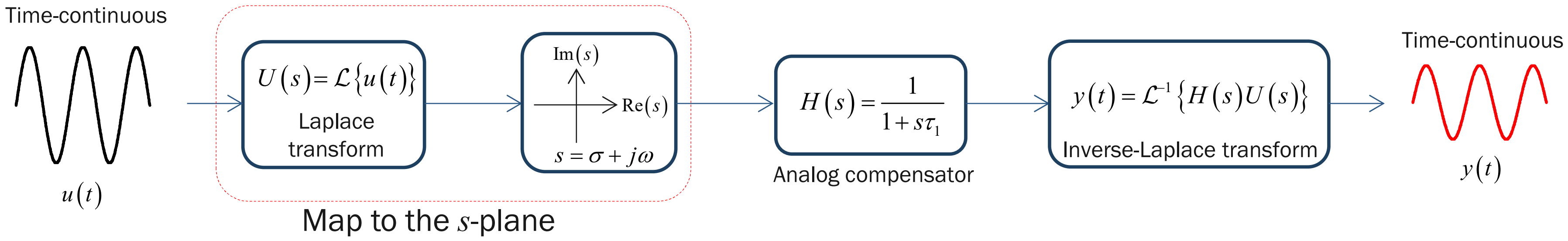
z-domain

$$U(z)=1+z^{-1}+z^{-2}+z^{-3}+\ldots$$

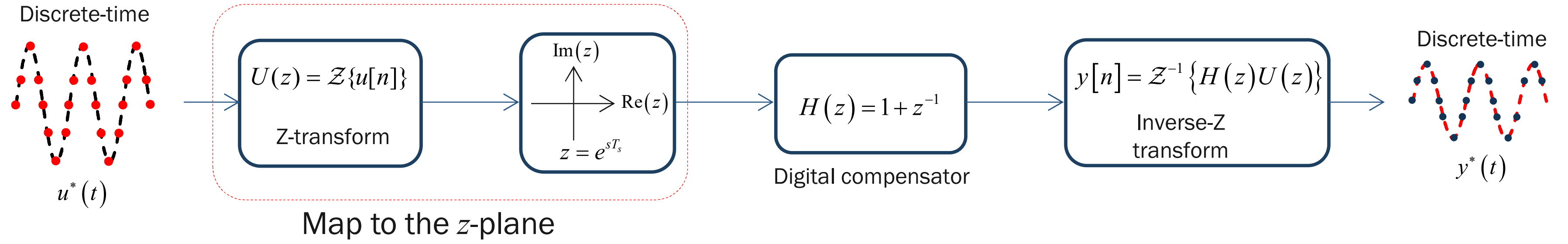


Laplace and z-Transforms

- ❑ Convert a continuous-time signal into a complex frequency function

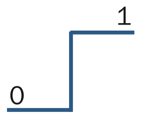


- ❑ Convert a discretized signal into a complex frequency function



Step-Response via Laplace- and z-Transforms Approaches

- The stimulus is continuous in time



$u(t)$

Time-continuous

step

$$\frac{1}{s}$$

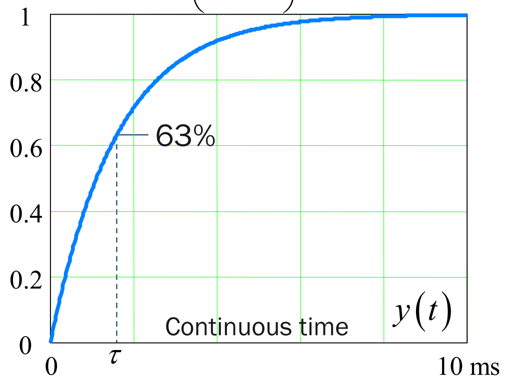
s-domain

$$H(s)=\frac{1}{1+s\tau_1}$$

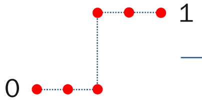
Transfer function in s

$$y(t)=\mathcal{L}^{-1}\left\{\frac{1}{s}\frac{1}{1+s\tau}\right\}$$

$$y(t)=u(t)\left(1-e^{-\frac{t}{\tau}}\right)$$



- The stimulus is a discretized signal



$u^*(t)$

Discrete time

step

$$\frac{z}{z-1}$$

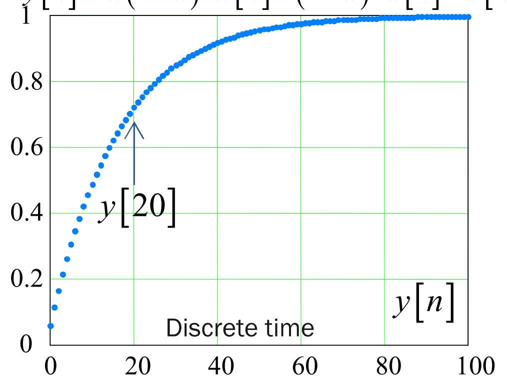
z-domain

$$H(z)=\frac{a}{1-(1-a)z^{-1}}$$

Transfer function in z

$$y[n]=\mathcal{Z}^{-1}\left\{\frac{z}{z-1}\frac{a}{1-(1-a)z^{-1}}\right\}$$

$$y[n]=a(1-a)^n u[n]-(1-a)^n u[n]+u[n]$$



From Laplace to z-Transform

- Take the Laplace transform of the sampled signal

$$U^*(s)=\mathcal{L}\{u^*(t)\}=\sum_{n=0}^{\infty}u[nT_s]e^{-snT_s}=u[0]+u[T_s]e^{-sT_s}+u[2T_s]e^{-2sT_s}+\ldots+u[nT_s]e^{-nsT_s}$$

- We can introduce the variable z and remove the exponentiel terms

$$\boxed{z=e^{sT_s}}\ \text{or}\ \boxed{z^{-1}=e^{-sT_s}}\longleftarrow$$

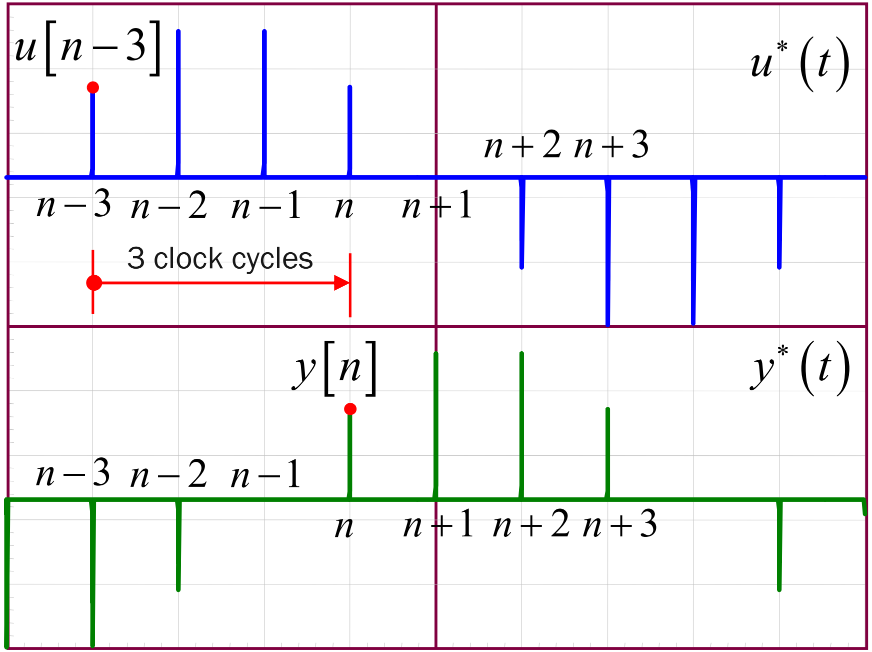
This is a delay

$$\Rightarrow\quad U(z)=\sum_{n=0}^{\infty}u[nT_s]z^{-n}=u[0]+u[T_s]z^{-1}+u[2T_s]z^{-2}+\ldots+u[nT_s]z^{-n}$$

- This is the definition of the unilateral z-transform

Z-Transform of a Time Delay

- ❑ The signal $y^*(t)$ is $u^*(t)$ shifted by 3 clock cycles



$$y[n] = u[n-k] \quad k = 3$$

$$\Downarrow$$

$$Y(z) = \mathcal{Z}\{u[n-k]\} = \sum_{n=0}^{\infty} u[n-k] \cdot z^{-n}$$

$$\boxed{Y(z) = U(z) \cdot z^{-k}}$$

$$\Downarrow$$

$$u[n] \rightarrow \boxed{z^{-3}} \rightarrow u[n-3]$$

$$\Downarrow$$

$$u[n] \rightarrow \boxed{z^{-1}} \rightarrow \boxed{z^{-1}} \rightarrow \boxed{z^{-1}} \rightarrow u[n-3]$$

Assemble Blocks to Form a Transfer Function

- Assume the following difference equation

$$y[n] = u[n] + u[n-1]$$

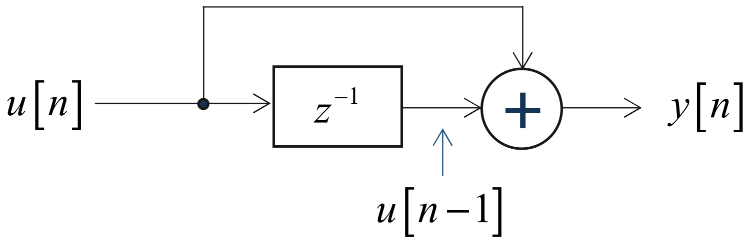
- Involve the delay operator to z-transform the expression

$$Y(z) = U(z) + U(z)z^{-1}$$

$$y[n-k] \rightarrow Y(z)\cdot z^{-k}$$
$$y[n] \rightarrow Y(z)$$
$$y[n+k] \rightarrow Y(z)\cdot z^{k}$$

- Factor and rearrange: $\dfrac{Y(z)}{U(z)} = 1 + z^{-1}$

- Describe this equation in a flow-graph style

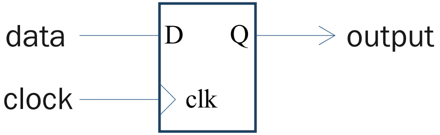


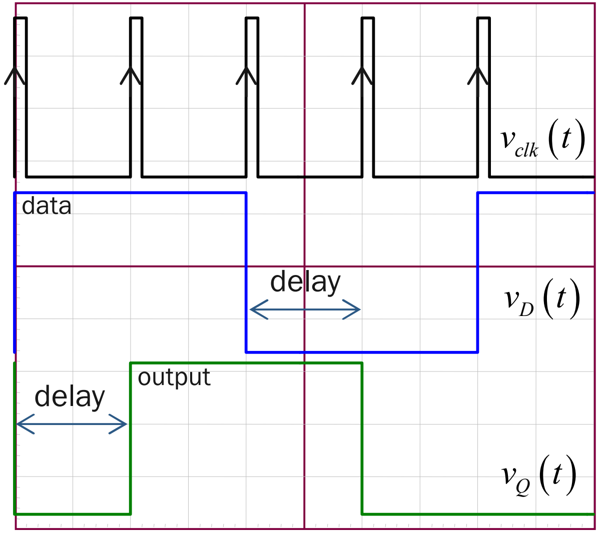
This is a simple low-pass filter

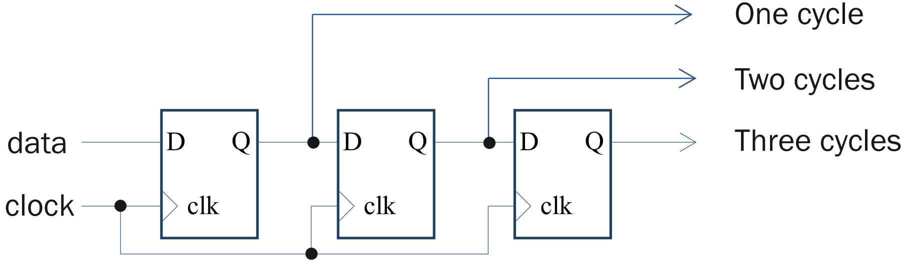
$$G(s) = G_0 \frac{1}{1+\dfrac{s}{\omega_p}} \qquad G_0 = 2 \qquad \omega_p = \frac{2}{T_s}$$

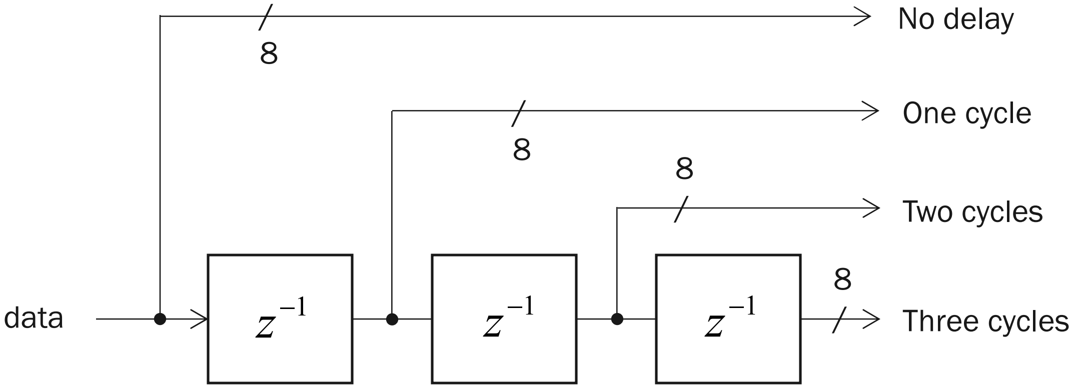
How do you Practically Build a Delay?

- A register delays the information by one clock cycle



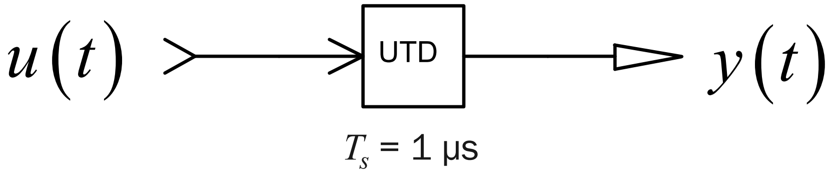






Modeling the Delay with SPICE

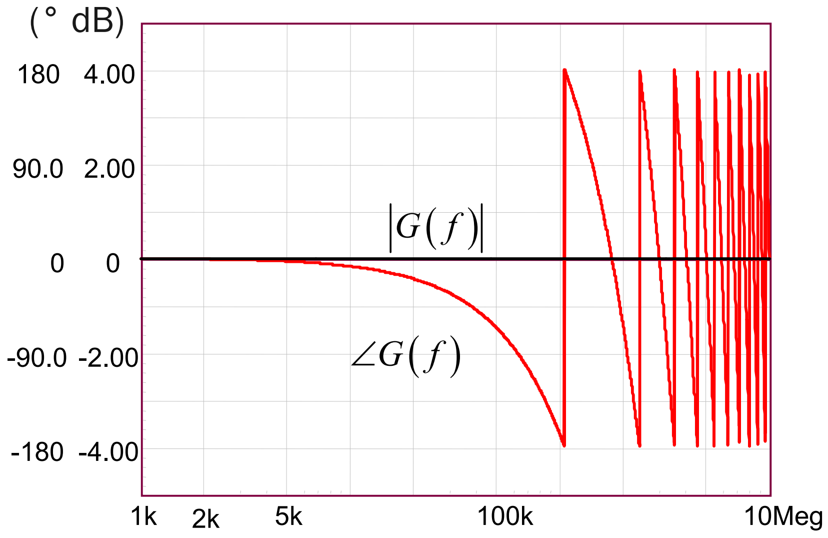
- A delay line in SPICE can be used to model a delay

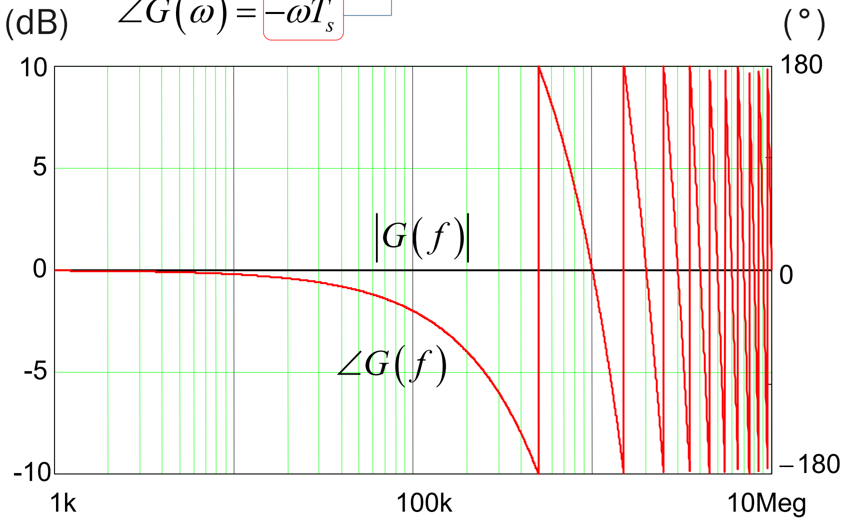


$$G(s) = e^{-sT_s} \equiv A e^{j\varphi}$$

$$|G(\omega)| = 1$$

$$\angle G(\omega) = -\omega T_s$$





S. Ben-Yaakov, D. Adar, Generic Average Modeling and Simulation of Discrete Controllers, APEC 2001

Approximating the Delay (1)

- A delay can be defined by associating a RHP zero and a LHP pole (all-pass filter)

$$G_d(s) = \frac{1 - \frac{s}{\omega_\tau}}{1 + \frac{s}{\omega_\tau}} \Rightarrow \quad |G_d(\omega)| = 1$$

Pole/zero cancel each other

$$\angle G_d = -\tan^{-1}\left(\frac{\omega}{\omega_\tau}\right) - \tan^{-1}\left(\frac{\omega}{\omega_\tau}\right) = -2\tan^{-1}\left(\frac{\omega}{\omega_\tau}\right)$$

- What value of τ satisfies this equality?

$$\arg\left(e^{-sT_s}\right) = \arg\left(\frac{1 - s/\omega_\tau}{1 + s/\omega_\tau}\right) \Rightarrow -\omega T_s = -2\tan^{-1}\left(\frac{\omega}{\omega_\tau}\right)$$

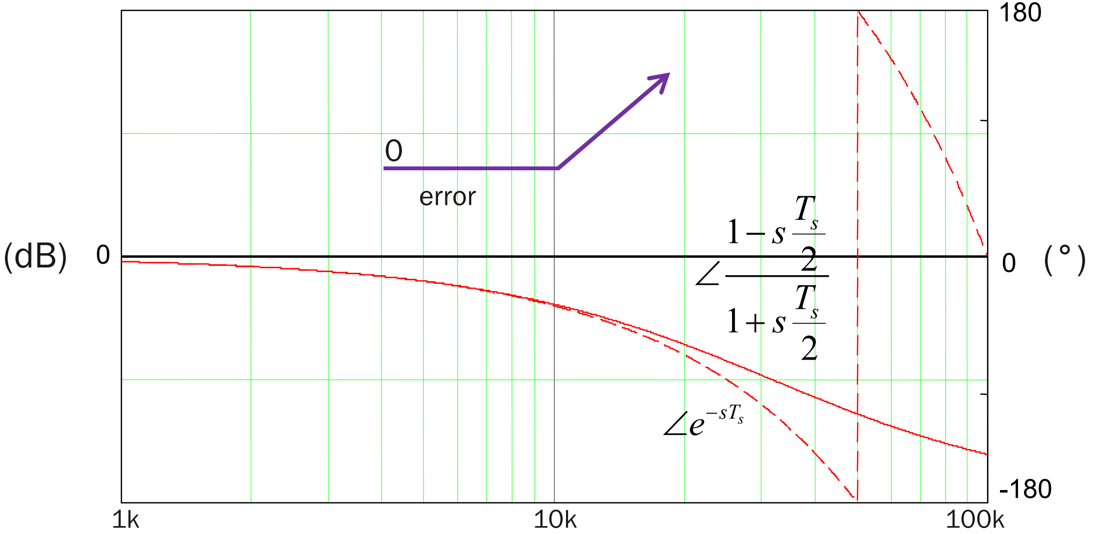
- Use the arctangent Taylor series equivalent:

$$\tan^{-1} x \approx x - \frac{x^3}{3} + \frac{x^5}{5} + \ldots \Rightarrow -\omega\tau \approx -2\left[\frac{\omega}{\omega_\tau} - \underbrace{\frac{\left(\frac{\omega}{\omega_\tau}\right)^3}{3} + \frac{\left(\frac{\omega}{\omega_\tau}\right)^5}{5}}_{\approx 0}\right]$$

Approximating the Delay (2)

- Solving for $\omega\tau$ gives us the 1st-order Padé approximant of the exponential

$$e^{-sT_s} \approx \frac{1 - s\frac{T_s}{2}}{1 + s\frac{T_s}{2}} = \frac{1 - \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}} \quad \text{with} \quad \omega_z = \omega_p = \frac{2}{T_s}$$



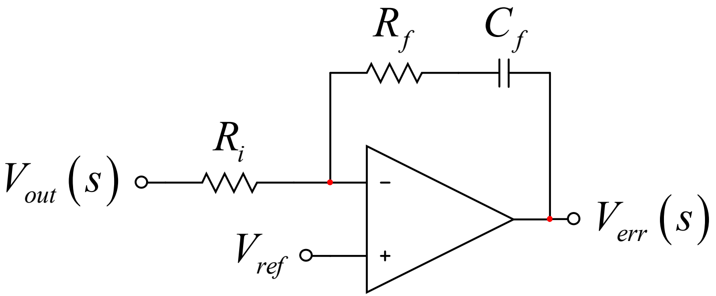
The phase deviation between both formulas increases with frequency

Agenda

- ❑ Introduction to Control Systems
- ❑ Compensation Strategies with an Operational Amplifier
- ❑ The TL431 at Work in Compensators
- ❑ Building a PID
- ❑ Continuous-Time to Discrete-Time Domains
- ❑ **Mapping and Compression**
- ❑ Digital Compensation
- ❑ Practical Implementation of Filters
- ❑ Conclusion

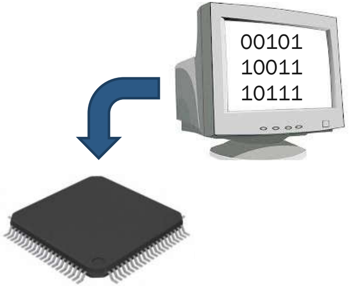
Discretizing Continuous-Time Transfer Functions

- ❑ How to move from one domain (continuous time) to the other one (discrete time)?



?

$$s = f(z)$$



Digital control

$$sV_{err}(s) = -s\frac{R_f}{R_i}V_{out}(s) - \frac{1}{R_iC_f}V_{out}(s)$$

$$b_0V_{err}(z) + b_1z^{-1}V_{err}(z) = a_0V_{out}(z) + a_1z^{-1}V_{err}(z)$$

$$z = e^{sT_s}$$

$$V_{out}(s) \rightarrow \boxed{G(s) = -G_0\left(1 + \frac{\omega_z}{s}\right)} \rightarrow V_{err}(s)$$

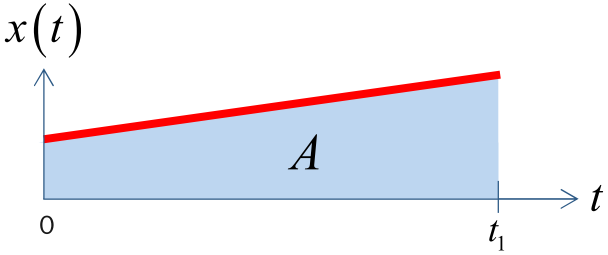
$$V_{out}^{\ *} \rightarrow \boxed{G(z) = \frac{a_0 + a_1z^{-1}}{b_0 + b_1z^{-1}}} \rightarrow V_{err}^{\ *}$$

Mapping s in the z-domain

- Assume a simple integrator transfer function

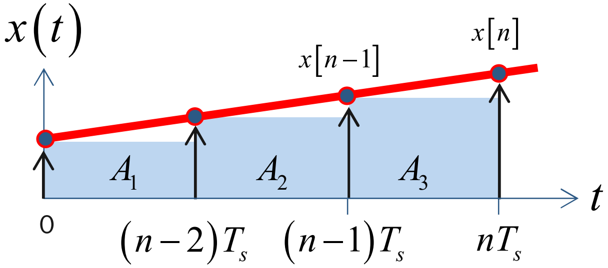
$$Y(s) = \frac{X(s)}{\tau s} \quad \Rightarrow \quad y(t) = \frac{1}{\tau}\int_0^t x(t)\,dt$$

- In continuous-time domain, the integral is the area under the curve



$$A = \int_0^{t_1} x(t)\,dt$$

➢ How do we evaluate the integral in the discrete-time domain?

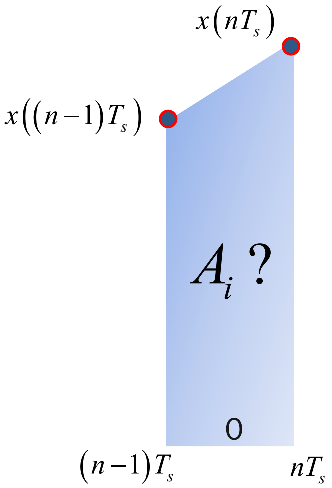


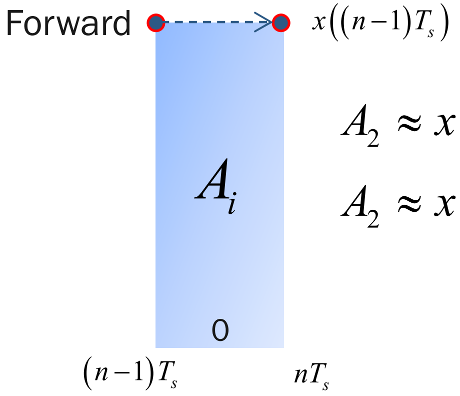
Sum discrete areas

$$A = A_1 + A_2 + A_3$$

Three Integration Options

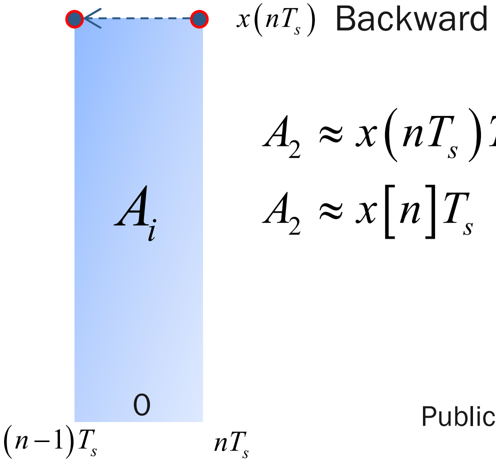
- The area can be approximated in three ways





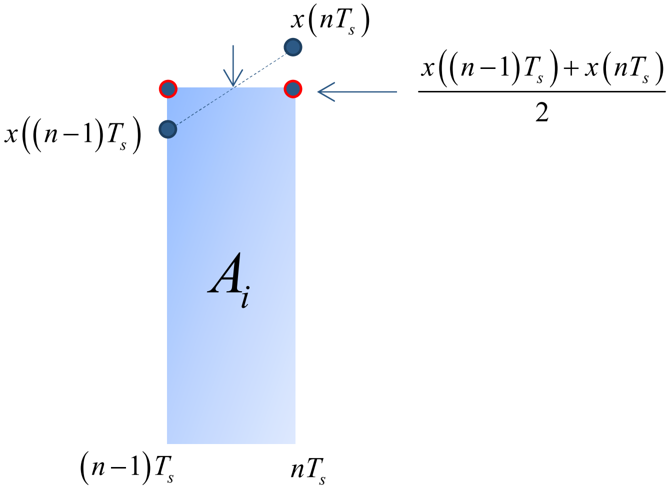
$$A_2 \approx x((n-1)T_s)T_s$$

$$A_2 \approx x[n-1]T_s$$



$$A_2 \approx x(nT_s)T_s$$

$$A_2 \approx x[n]T_s$$



$$A_2 \approx \frac{x((n-1)T_s)+x(nT_s)}{2}T_s$$

$$A_2 \approx \frac{x[n-1]+x[n]}{2}T_s$$

- Trapezoidal integration
- Flat-top approximation

Discretizing in Different Ways

Forward-Euler method

$$y[n] = y[n-1] + x[n-1]T_s$$

Integral value at sample n — Integral value at previous sample — Approximate integral between samples

$$Y(z) = Y(z)z^{-1} + X(z)z^{-1}T_s$$

$$\frac{Y(z)}{X(z)} = \frac{z^{-1}T_s}{1-z^{-1}} \times \frac{z}{z} = \frac{T_s}{z-1} \Leftrightarrow \frac{Y(s)}{X(s)} = \frac{1}{s}$$

$$\Rightarrow H(z) = H(s)\Big|_{s=\frac{z-1}{T_s}} \qquad \boxed{s = \frac{z-1}{T_s}}$$

Backward-Euler method

$$y[n] = y[n-1] + x[n]T_s$$

Integral value at sample n — Integral value at previous sample — Approximate integral between samples

$$Y(z) = Y(z)z^{-1} + X(z)T_s$$

$$\frac{Y(z)}{X(z)} = \frac{T_s}{1-z^{-1}} \times \frac{z}{z} = \frac{zT_s}{z-1} \Leftrightarrow \frac{Y(s)}{X(s)} = \frac{1}{s}$$

$$\Rightarrow H(z) = H(s)\Big|_{s=\frac{z-1}{zT_s}} \qquad \boxed{s = \frac{z-1}{zT_s}}$$

Tustin and Backward/Forward Euler Methods

❑ Tustin method

$$y[n] = y[n-1] + \frac{x[n-1]+x[n]}{2}T_s$$

Integral value at sample n

Integral value at previous sample

Approximate integral between samples

$$2Y(z) = 2Y(z)z^{-1} + X(z)z^{-1}T_s + X(z)T_s$$

$$2Y(z)\left(1-z^{-1}\right) = X(z)T_s\left(1+z^{-1}\right)$$

$$Y(z) = X(z)\frac{T_s}{2}\frac{\left(1+z^{-1}\right)}{\left(1-z^{-1}\right)}\times\frac{z}{z} \Leftrightarrow \frac{Y(s)}{X(s)} = \frac{1}{s}$$

$$\Rightarrow H(z) = H(s)\Big|_{s=\frac{2}{T_s}\frac{z-1}{z+1}} \qquad \boxed{s = \frac{2}{T_s}\frac{z-1}{z+1}}$$

❑ Other approaches

Discretization Method	What is Important?
Zero-Order Hold (ZOH)	Transient response with a staircase input
First-Order Hold (FOH)	Transient response with a PWL input
Impulse-Invariant Mapping	Transient response with impulse train input
Pole-Zero Matching	Good matching between frequency- and discrete domains

✓ Tustin brings a good matching between frequency- and discrete-domains

https://www.mathworks.com/help/control/ug/continuous-discrete-conversion-methods.html

Viewing Tustin, Backward- and Direct-Euler Differently

- It is possible to approximate the exponential using Taylor's series expansion

$$e^x \approx 1 + x + x^2 + \ldots + \frac{1}{(n-1)!}x^{n-1} \quad \Rightarrow \quad e^x \approx 1 + x$$

Neglecting
high-order terms

✓ Forward-Euler

$$z = e^{sT_s} \approx 1 + sT_s \quad \Rightarrow \quad \boxed{s = \frac{z-1}{T_s}}$$

✓ Backward-Euler

$$z = \frac{1}{e^{-sT_s}} \approx \frac{1}{1 - sT_s} \quad \Rightarrow \quad \boxed{s = \frac{z-1}{zT_s}}$$

✓ Tustin transform

$$\boxed{s = \frac{2}{T_s}\frac{z-1}{z+1}} \quad \Leftarrow \quad z \approx \frac{1 + s\frac{T_s}{2}}{1 - s\frac{T_s}{2}}$$

Use the Padé
approximant of
the delay operator

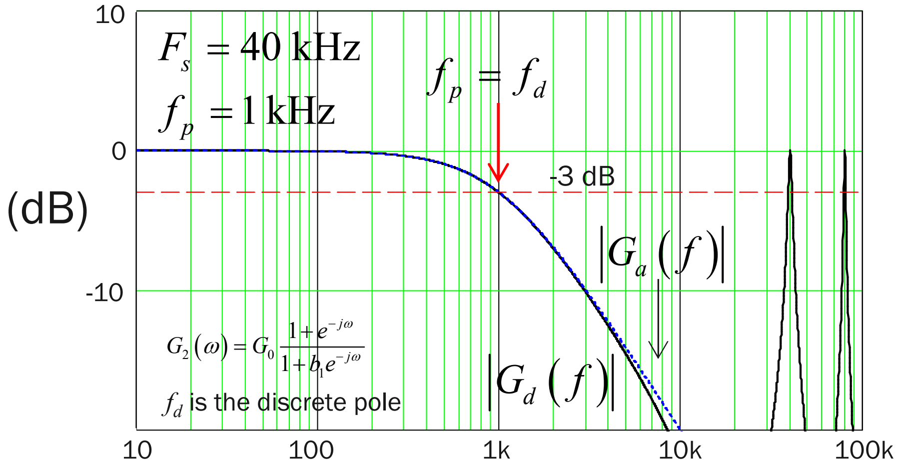
Henri Padé
1863-1953

Applying Tustin's Method to a Low-Pass Filter

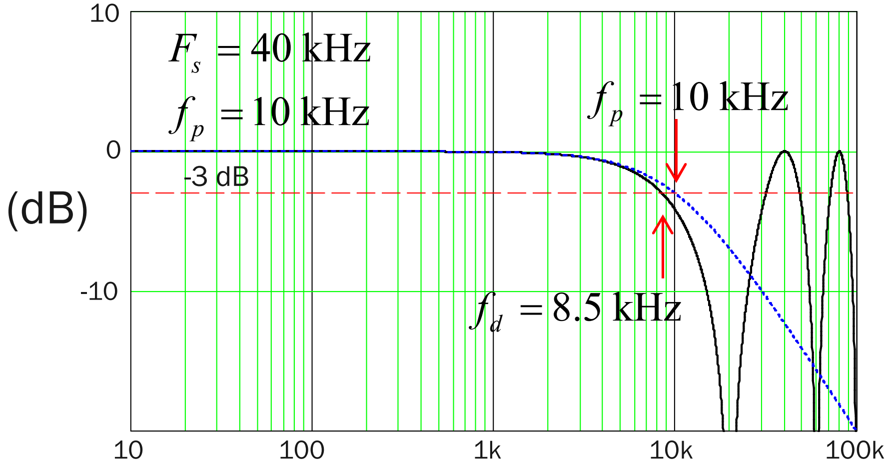
- We have the following low-pass filter transfer function

$$G_a(s) = \frac{1}{1+\frac{\boxed{s}}{\omega_p}} \quad s = \frac{2}{T_s}\frac{z-1}{z+1} \quad \Rightarrow \quad G_d(z) = G_0\frac{1+z^{-1}}{1+b_1\boxed{z^{-1}}} \quad G_0 = 1-\frac{2}{T_s\omega_p+2} \quad b_1 = \frac{T_s\omega_p-2}{T_s\omega_p+2}$$

- Test the response in the frequency domain with $z^{-1} = e^{-sT_s}$



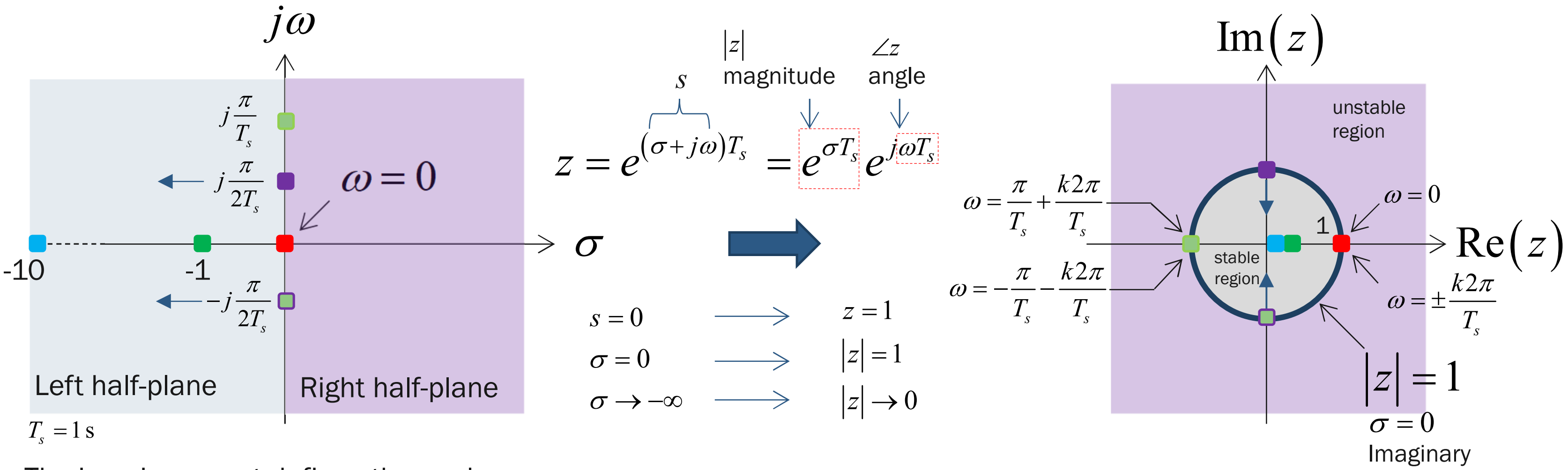
$$\frac{f_p - f_d}{f_p} = 0.2\%$$



$$\frac{f_p - f_d}{f_p} = 15.2\%$$

From the s-Plane to the z-Plane – Exact Mapping

- ❑ The stable pole region – left half-plane – entirely maps inside the unit circle



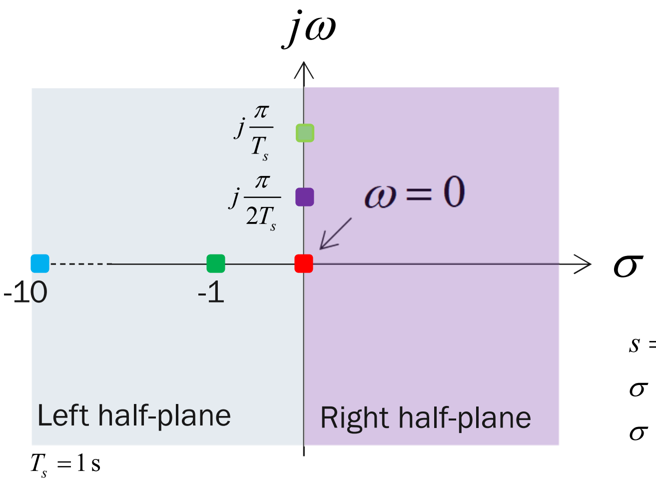
The imaginary part defines the angle
The real part defines the radius

- ➢ The stable poles (LHPP) in s are confined inside the unit circle
- ➢ Unstable poles (RHPP) in s lie outside the unit circle

Watch Brian Douglas's videos at https://www.youtube.com/channel/UCq0imsn84ShAe9PBOFnoIrg

The Bilinear Transform

- Tustin's approximation maps the entire $j\omega$ axis on the unit circle in the z-plane

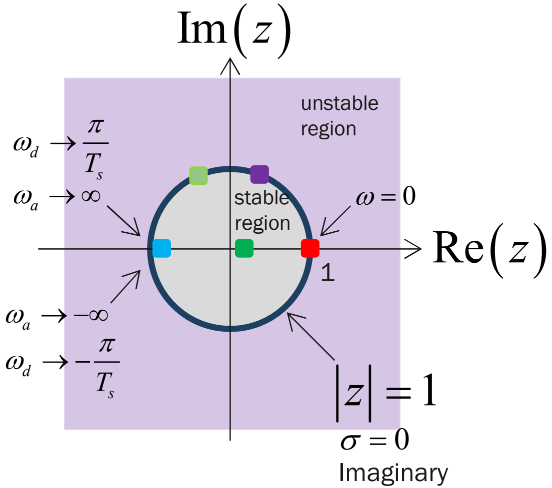


$$z = \frac{1 + s\frac{T_s}{2}}{1 - s\frac{T_s}{2}}$$

$s = 0 \longrightarrow z = 1$

$\sigma = 0 \longrightarrow |z| = 1$

$\sigma \to -\infty \longrightarrow |z| \to 1$



- The stable poles (LHPP) in s are confined inside the unit circle
- Unstable poles (RHPP) in s lie outside the unit circle

Stable in the s-domain ⟺ Stable in the z-domain

ω_a = analog world

ω_d = discrete world

Compression in the Frequency Response

- Map the s-domain transfer function in the z-domain with bilinear transform

$$H_a(s)=\frac{1}{1+s\tau}\xrightarrow{\text{Map to } z} H_a(z)=\frac{1}{1+\left(\frac{2}{T_s}\frac{z-1}{z+1}\right)\tau}$$

- In general:

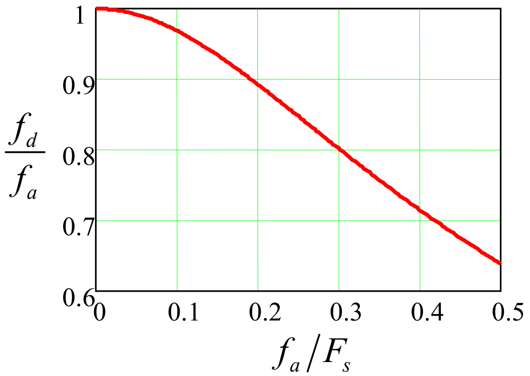
$$z=e^{j\omega_d T_s}$$

$$\underset{\text{discrete}}{H_d}(z)=\underset{\text{analogue}}{H_a}\left(\frac{2}{T_s}\frac{z-1}{z+1}\right) \Rightarrow H_d\left(e^{j\omega_d T_s}\right)=H_a\left(\frac{2}{T_s}\frac{e^{j\omega_d T_s}-1}{e^{j\omega_d T_s}+1}\right)\overset{\text{Euler}}{=}H_a\left(j\frac{2}{T_s}\tan\left(\frac{\omega_d T_s}{2}\right)\right)$$

- The discretized pole ω_d is not at its original frequency-domain position

$$\underset{\text{discrete}}{\omega_d}=\frac{2}{T_s}\tan^{-1}\left(\frac{\overset{\text{analogue}}{\omega_a} T_s}{2}\right)$$

Frequency warping



What you have → f_d; What is expected → f_a

$$f_d=\frac{1}{\pi T_s}\tan^{-1}\left(f_a\pi T_s\right)$$

$$f_a=1\text{ kHz}\longrightarrow f_d=0.997\text{ kHz}$$

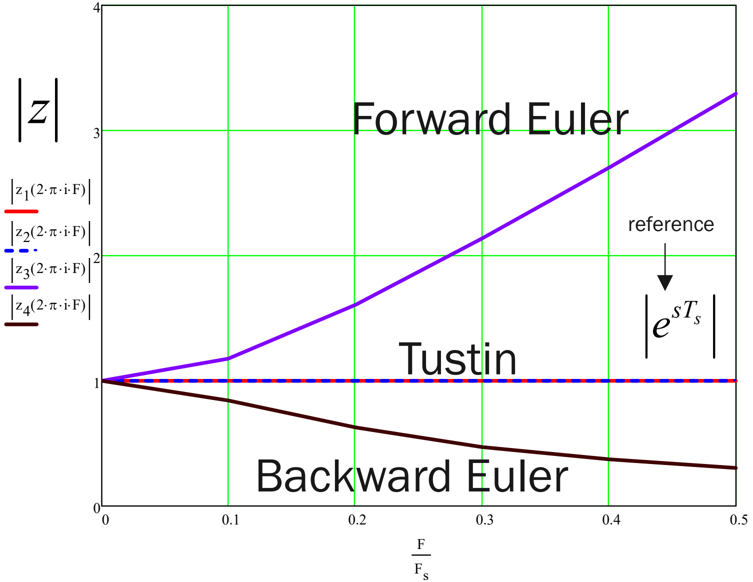
discretized

$$f_a=10\text{ kHz}\longrightarrow f_d\approx 8.5\text{ kHz}$$

$$F_s=40\text{ kHz}$$

Magnitude and Phase Distortion of Mappings

- ❑ Tustin offers the best magnitude tracking while phase diverges as you approach $\frac{F_s}{2}$

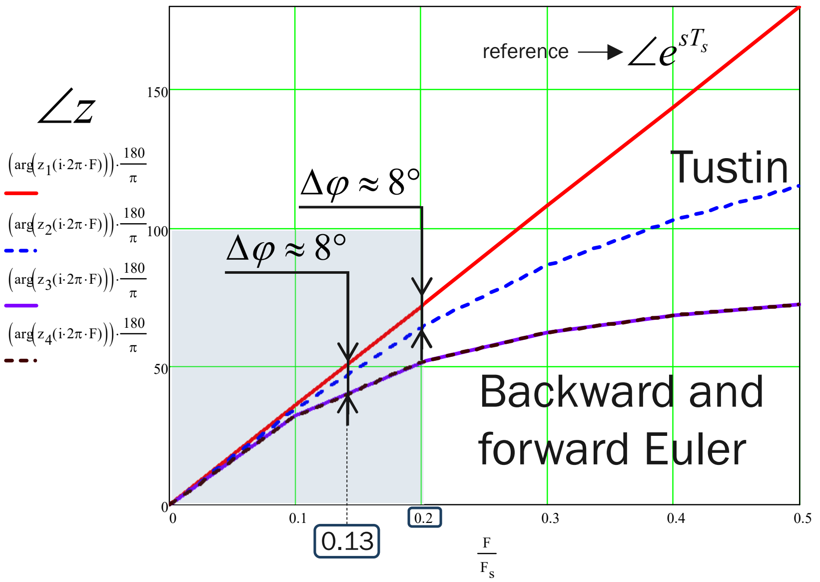


Exact
$$z_1(s) := e^{s \cdot T_s}$$

Tustin
$$z_2(s) := \frac{1 + s \cdot \frac{T_s}{2}}{1 - s \cdot \frac{T_s}{2}}$$

Forward
$$z_3(s) := 1 + s \cdot T_s$$

Backward
$$z_4(s) := \frac{1}{1 - s \cdot T_s}$$

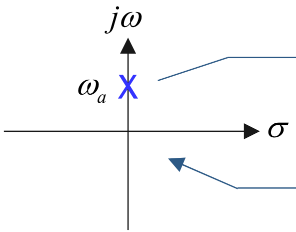


- ❑ Tustin provides same phase-mapping accuracy with lower sampling frequency
  - ➢ Acceptable results for $F_s > 5 \cdot f$ while Euler would require $F_s > 7 \cdot f$

M. Jovanović, *Introduction to Digital Control of Switch-Mode Converters*, in-house course, PHX 2014

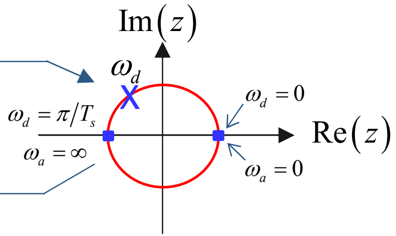
Account for Warping when Positioning Poles and Zeroes

- Frequency compression occurs because of nonlinear relationship linking ω_d and ω_a



$$\omega_d = \frac{2}{T}\tan^{-1}\left(\frac{\omega_a T}{2}\right)$$

$$\omega_a = \frac{2}{T}\tan\left(\frac{\omega_d T}{2}\right)$$



$$-\infty < \omega_a < +\infty$$

Entire continuous frequency range

$$-\frac{\pi}{T_s} < \omega_d < +\frac{\pi}{T_s}$$

Discrete-time frequency

- You can anticipate the compression and *prewarp* your poles and zeroes

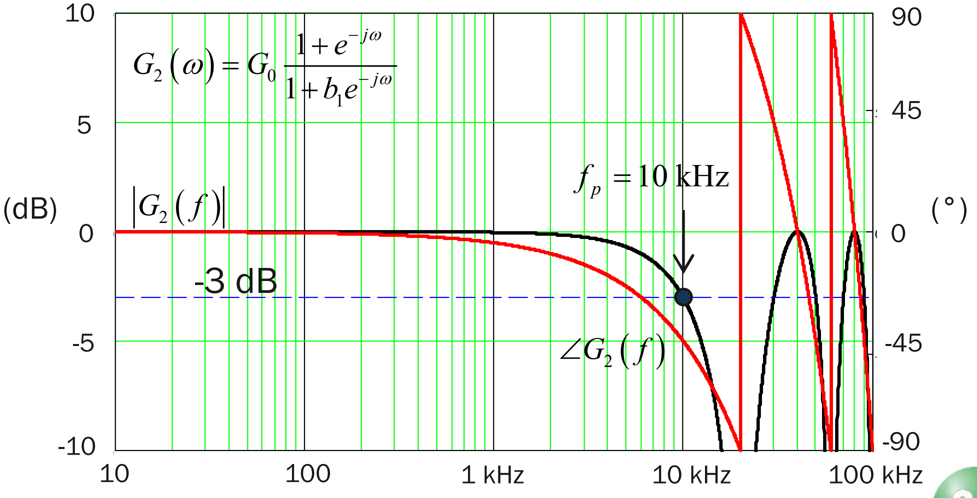
Where you should set the pole

What you want in discrete time

$$\omega_{p'} = \frac{2}{T_s}\tan\left(\frac{\omega_d T_s}{2}\right)$$

$$f_{p'} = \frac{2}{25u}\tan\left(\frac{2\pi \cdot 10k \cdot T_s}{2}\right)\frac{1}{2\pi} = 12.7\ \text{kHz}$$

$$G_0 = 1 - \frac{2}{T_s\omega_{p'} + 2} \qquad b_1 = \frac{T_s\omega_{p'} - 2}{T_s\omega_{p'} + 2}$$



Agenda

- ❑ Introduction to Control Systems
- ❑ Compensation Strategies with an Operational Amplifier
- ❑ The TL431 at Work in Compensators
- ❑ Building a PID
- ❑ Continuous-Time to Discrete-Time Domains
- ❑ Mapping and Compression
- ❑ **Digital Compensation**
- ❑ Practical Implementation of Filters
- ❑ Conclusion

From the z-Equation to Practical Implementation

- Start with the z-transform expression

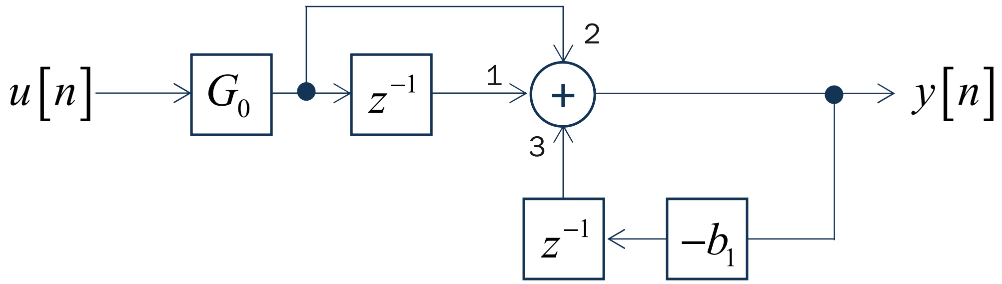
$$\frac{Y(z)}{U(z)} = G_0 \frac{1+z^{-1}}{1+b_1 z^{-1}} \xrightarrow{\text{Expand}} Y(z)+Y(z)b_1 z^{-1} = U(z)G_0 + U(z)G_0 z^{-1}$$

Rearrange

$$Y(z) = U(z)G_0 + U(z)G_0 z^{-1} - Y(z)b_1 z^{-1}$$

$$\Rightarrow y[n] = u[n]G_0 + u[n-1]G_0 - y[n-1]b_1$$

- Assemble blocks according to the obtained equation

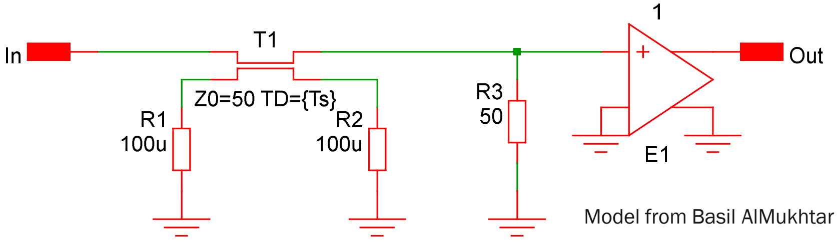


How can we test this configuration?

Public Information

SPICE Simulates Delays Efficiently

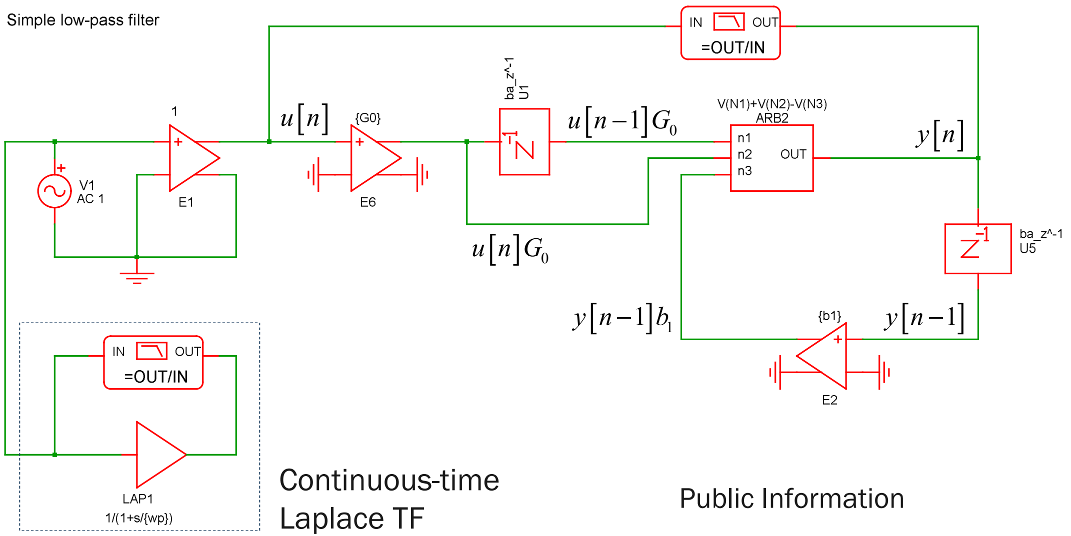
- We know z^{-1} can be modeled by a delay line in SIMextrix®



Model from Basil AlMukhtar – ON Semiconductor

- By reproducing the flow-graph blocks, you can build the transfer function

Simple low-pass filter

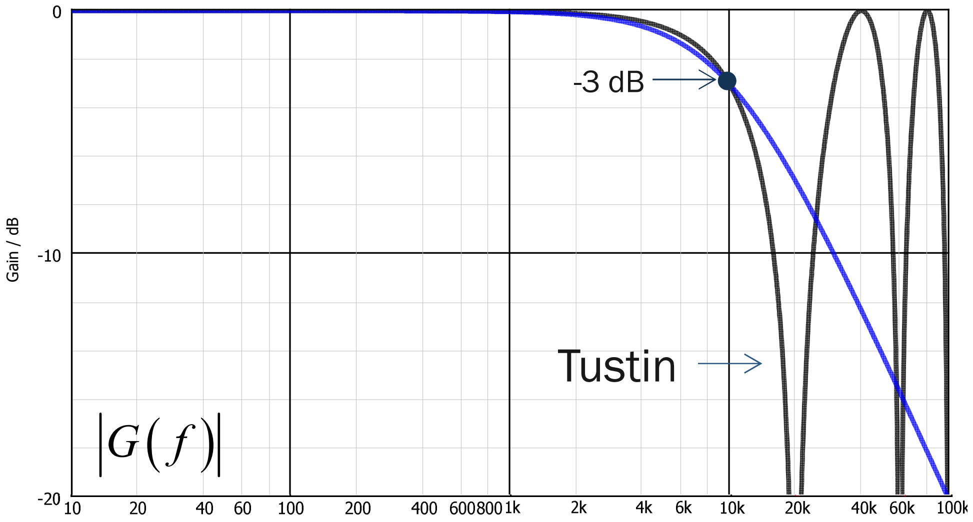


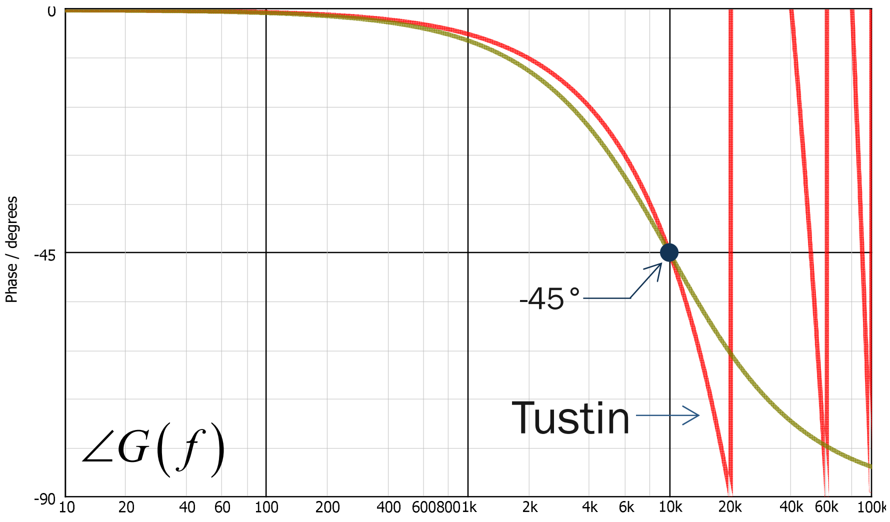
Continuous-time
Laplace TF

```
*
.param Fs=40k
.param Ts={1/Fs}
.param fp=10k
.param wp={2*pi*fp}
.param wpw={(2/Ts)*tan(wp*Ts/2)}
.param G0={1-2/(Ts*wpw+2)}
.param b1={(Ts*wpw-2)/(Ts*wpw+2)}
*
```

Plot the Continuous-Time Response

- ❑ Pole frequency pre-warping offers an excellent matching





- ➢ Phase extends beyond -90° with discrete-time implementation
- ✓ Flowchart structure and behavior is validated before coding begins

SIMPLIS® Include 1st -Order Filters Blocks

- A pole-zero equation implemented in SIMPLIS® is the following:

$$T(z)=\frac{N_1+N_0z^{-1}}{1+D_0z^{-1}}=N_1\frac{1+\frac{N_0}{N_1}z^{-1}}{1+D_0z^{-1}}$$

Expand

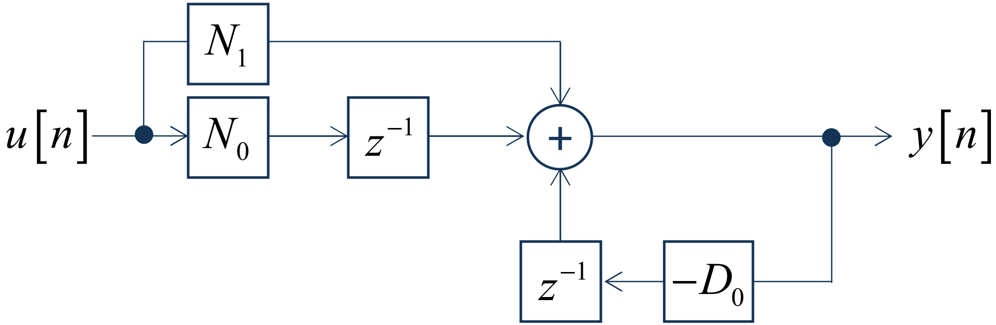
$$Y(z)+Y(z)D_0z^{-1}=U(z)N_1+U(z)N_0z^{-1}$$

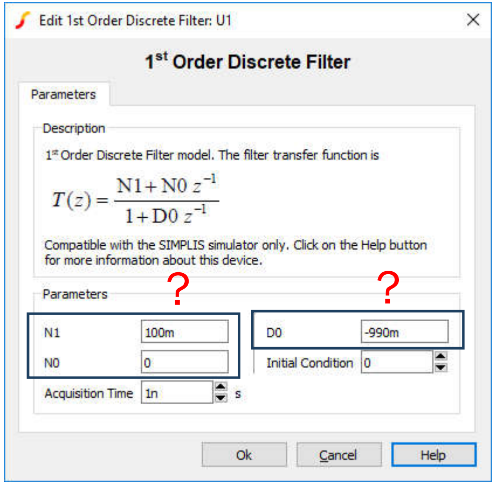
Rearrange

$$Y(z)=U(z)N_1+U(z)N_0z^{-1}-Y(z)D_0z^{-1}$$

$$y[n]=u[n]N_1+u[n-1]N_0-y[n-1]D_0$$

- Assemble blocks according to the obtained equation





Determining Coefficients Values

- Replace z by its value in the s-domain: $z^{-1} = e^{-sT_s}$

$$H(z) = \frac{N_1 + N_0 \boxed{z^{-1}}}{1 + D_0 \boxed{z^{-1}}} \Rightarrow H(s) = \frac{N_1 + N_0 \boxed{e^{-sT_{sw}}}}{1 + D_0 e^{-sT_{sw}}} \leftarrow e^{-sT_{sw}} \approx \frac{1 - s\frac{T_{sw}}{2}}{1 + s\frac{T_{sw}}{2}}$$

If we plug this expression into the original transfer function, we have:

$$H(s) \approx \frac{N_1 + N_0 \dfrac{1 - s\frac{T_{sw}}{2}}{1 + s\frac{T_{sw}}{2}}}{1 + D_0 \dfrac{1 - s\frac{T_{sw}}{2}}{1 + s\frac{T_{sw}}{2}}} \xrightarrow{\text{rearrange}} H(s) = H_0 \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{\omega_p}}$$

$$\omega_z = \frac{2}{T_{sw}} \frac{N_1 + N_0}{N_1 - N_0}$$

$$\omega_p = \frac{2}{T_{sw}} \frac{1 + D_0}{1 - D_0}$$

$$H_0 = \frac{N_0 + N_1}{1 + D_0}$$

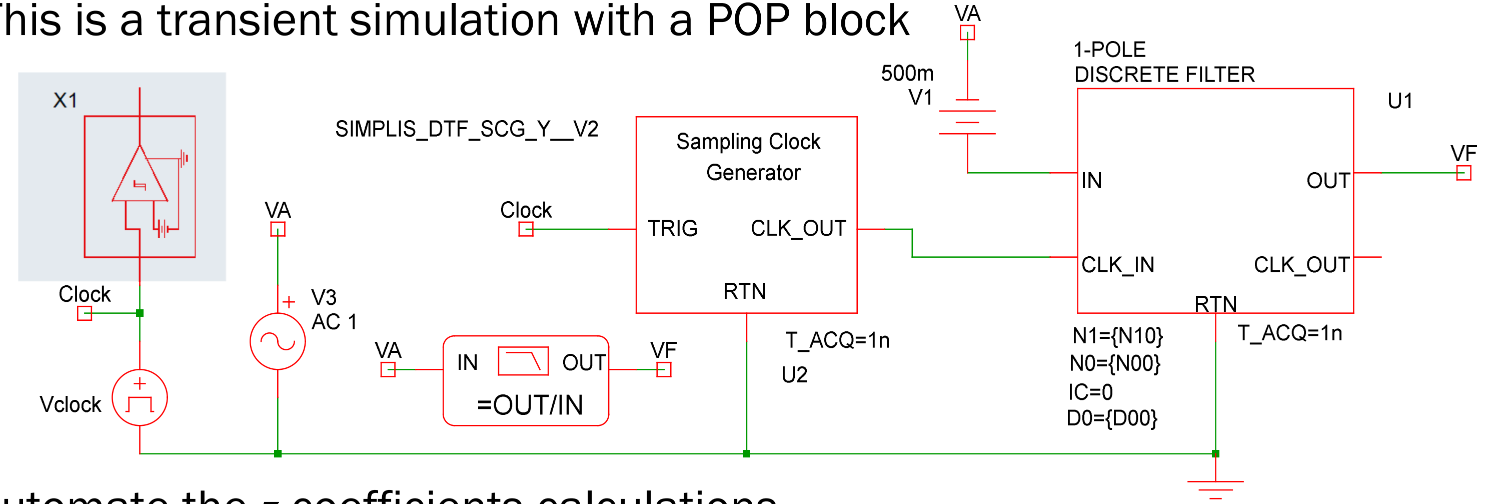
$$N_0 = \frac{H_0 \omega_p (T_s \omega_z - 2)}{\omega_z (T_s \omega_p + 2)}$$

$$N_1 = \frac{H_0 \omega_p (T_s \omega_z + 2)}{\omega_z (T_s \omega_p + 2)}$$

$$D_0 = 1 - \frac{4}{T_s \omega_p + 2}$$

The Implementation Requires a Clock Generator

- This is a transient simulation with a POP block



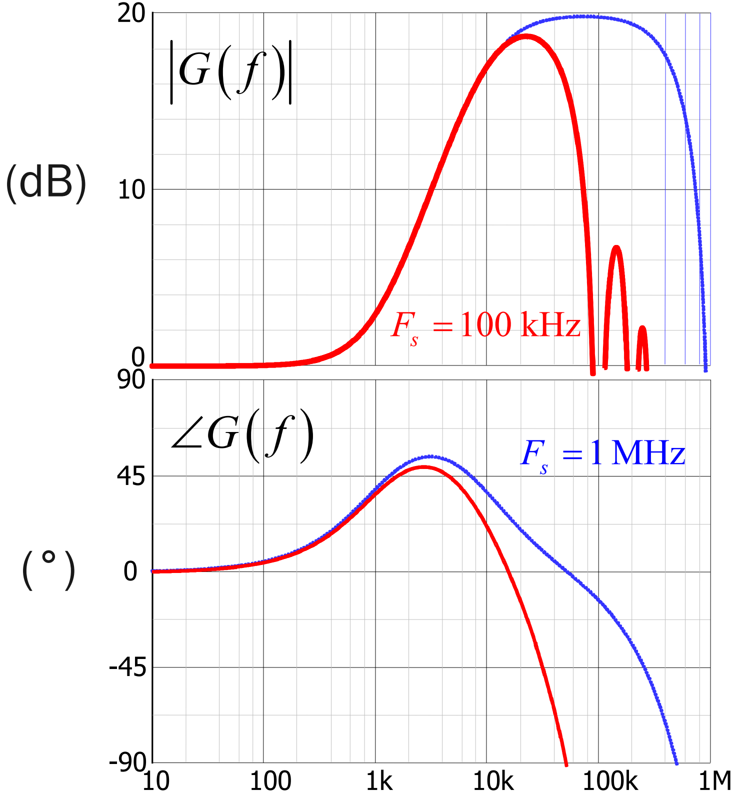
- Automate the z-coefficients calculations

```
*
.GLOBALVAR fp=10k * pole position *
.GLOBALVAR fz=1k * zero position *
.GLOBALVAR H0=0 * dc gain in dB *
.GLOBALVAR Fsw=100k * sampling frequency *
*
.GLOBALVAR Tsw=1/Fsw
.GLOBALVAR H00=10^(H0/20)
.GLOBALVAR wp=2*pi*fp
.GLOBALVAR wz=2*pi*fz
```

```
*
.GLOBALVAR N10=H00*wp*(Tsw*wz+2)/(wz*(Tsw*wp+2))
.GLOBALVAR N00=H00*wp*(Tsw*wz-2)/(wz*(Tsw*wp+2))
.GLOBALVAR D00=1-4/(Tsw*wp+2)
* .GLOBALVAR N00=0
* if no zero, set N00= 0 and update N10 with below line *
* .GLOBALVAR N10=H00*(wp*Tsw)/(1+(Tsw*wp/2))
*
```

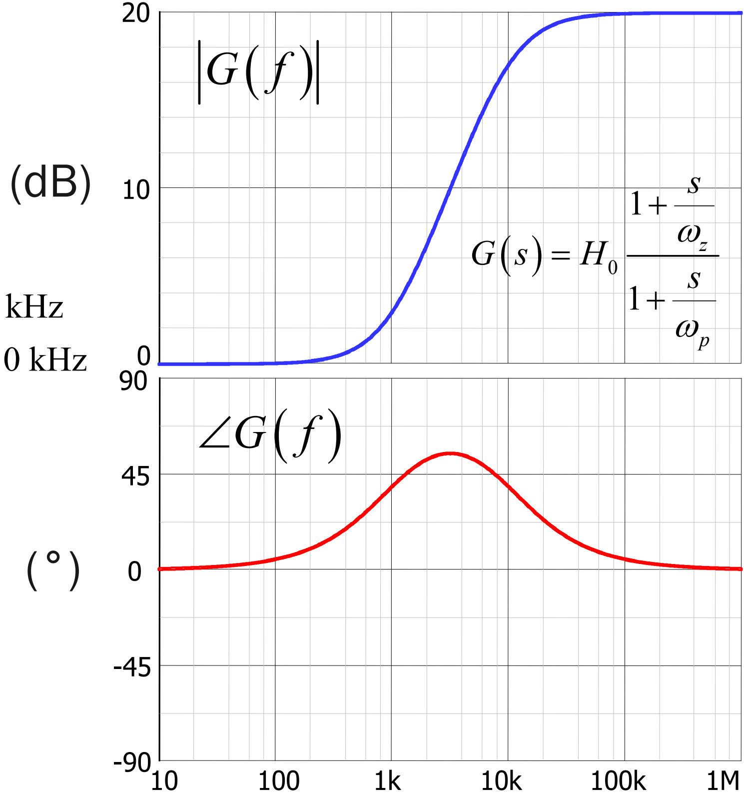
Sampling Frequency Affects the Filter Response

- ❑ The phase drops faster at a 100-kHz sampling frequency



$f_z = 1\ \text{kHz}$

$f_p = 10\ \text{kHz}$



Agenda

- ❑ Introduction to Control Systems
- ❑ Compensation Strategies with an Operational Amplifier
- ❑ The TL431 at Work in Compensators
- ❑ Building a PID
- ❑ Continuous-Time to Discrete-Time Domains
- ❑ Mapping and Compression
- ❑ Digital Compensation
- ❑ **Practical Implementation of Filters**
- ❑ Conclusion

Implementing a Type 2 Compensator

- You start from the *low-entropy* form Laplace transfer function

$$G(s) = G_0 \frac{1+\frac{\omega_z}{s}}{1+\frac{s}{\omega_p}} \Rightarrow s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}} \Rightarrow G(z) = \frac{G_0 T_s \omega_p (1+z)(2z + T_s\omega_z + T_s\omega_z z - 2)}{4z^2 - 8z - 2T_s\omega_p + 2T_s\omega_p z^2 + 4}$$

Divide by z^2

Divide by z^2

Biquad filter

$$\Rightarrow G(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}}$$

$$a_0 = \frac{G_0 T_s \omega_p (T_s\omega_z + 2)}{2(2 + T_s\omega_p)}$$

$$a_1 = \frac{G_0 T_s^2 \omega_p \omega_z}{2 + T_s\omega_p} \qquad b_1 = -\frac{2}{1 + 0.5T_s\omega_p}$$

$$a_2 = \frac{G_0 T_s \omega_p (T_s\omega_z - 2)}{2T_s\omega_p + 4} \qquad b_2 = \frac{2}{1 + 0.5T_s\omega_p} - 1$$

Specifications:

$G_{f_c} = 20\text{ dB} \quad f_c = 1\text{ kHz}$

$Boost = 50°$

$k = \tan\left(\frac{boost}{2} + \frac{\pi}{4}\right) = 2.74$

$f_z = 364\text{ Hz}$

$f_p = 2.74\text{ kHz}$

$F_s = 1\text{ MHz}$

$a_0 = 0.0857$

$a_1 = 1.957 \cdot 10^{-4}$

$a_2 = \text{-}0.0855$

$b_1 = \text{-}1.9829$

$b_2 = 0.9829$

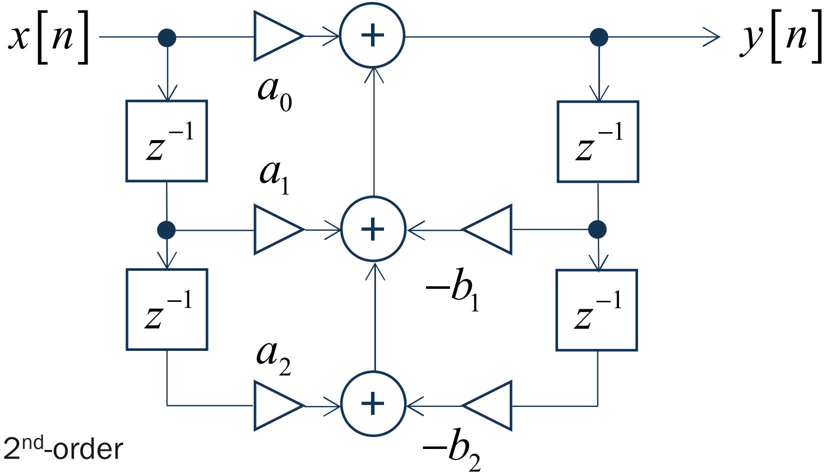
The Biquad Filter is a Recursive Digital Structure

- A part of the output is fed back to form the transfer function

Feedforward coefficients

Feedback coefficients

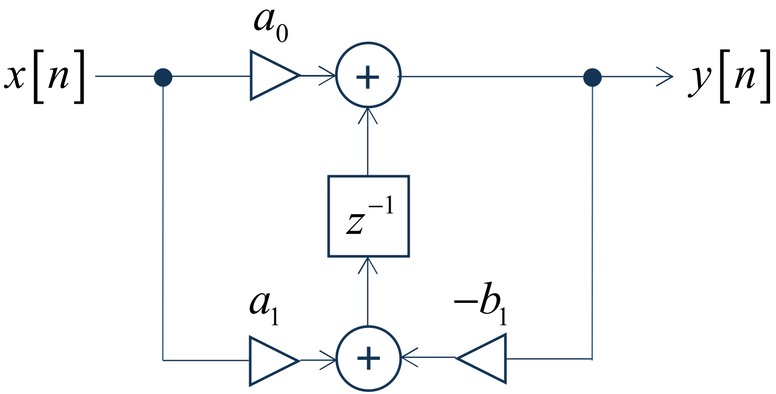
$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] - b_1 y[n-1] - b_2 y[n-2]$$



2nd-order

Reduce to 1st order filter

$a_2 = b_2 = 0$



- This configuration is called *Direct Implementation I*
  - Can be *transposed* in different forms (DI II) so as to minimize calculation errors

Where are Poles and Zeroes?

- If you read coefficients from the code or a circuit: where are poles and zeroes?

$$e^{-sT_{sw}} \approx \frac{1-s\dfrac{T_{sw}}{2}}{1+s\dfrac{T_{sw}}{2}}$$

$$G(z)=\frac{a_0+a_1z^{-1}+a_2z^{-2}}{1+b_1z^{-1}+b_2z^{-2}}$$

$G_0 \to \infty$

Double zero

$$G(s)\approx \boxed{\frac{a_0+a_1+a_2}{b_1+b_2+1}}\,\frac{1+sT_s\dfrac{a_0-a_2}{a_0+a_1+a_2}+s^2\left(\dfrac{T_s}{2}\right)^2\dfrac{a_0-a_1+a_2}{a_0+a_1+a_2}}{1+sT_s\dfrac{1-b_2}{b_1+b_2+1}+s^2\left(\dfrac{T_s}{2}\right)^2\dfrac{b_2-b_1+1}{b_1+b_2+1}}$$

Double pole

- Numerator and denominator follow the form:

$$N(s)=a_0+a_1+a_2+sT_{sw}(a_0-a_2)+s^2\left(\frac{T_s}{2}\right)^2(a_0-a_1+a_2)$$

$$N(s)=1+s(\tau_{z1}+\tau_{z2})+s^2\tau_{z1}\tau_{z2}$$

$$D(s)=1+b_1+b_2+sT_s(1-b_2)+s^2\left(\frac{T_s}{2}\right)^2(b_2-b_1+1)$$

$$D(s)=1+s(\tau_{p1}+\tau_{p2})+s^2\tau_{p1}\tau_{p2}$$

$$G(s)=G_0\frac{1+s(\tau_{z1}+\tau_{z2})+s^2\tau_{z1}\tau_{z2}}{1+s(\tau_{p1}+\tau_{p2})+s^2\tau_{p1}\tau_{p2}}$$

Normalized *low-entropy* form

C. Basso, *Linear Circuit Transfer Function: an Introduction to Fast Analytical Techniques, Wiley, IEEE-Press, 2016*

Link Time Constants and Coefficients

- Solve a system of equations to unveil the poles and zeroes

$$\tau_{z1}+\tau_{z2}=\frac{a_0-a_2}{a_0+a_1+a_2}T_s \qquad \tau_{z1}\cdot\tau_{z2}=\frac{a_0-a_1+a_2}{a_0+a_1+a_2}\left(\frac{T_s}{2}\right)^2$$

$$G_0=\frac{a_0+a_1+a_2}{1+b_1+b_2}$$

$$\tau_{p1}+\tau_{p2}=\frac{1-b_2}{1+b_1+b_2}T_s \qquad \tau_{p1}\cdot\tau_{p2}=\frac{b_2-b_1+1}{1+b_1+b_2}\left(\frac{T_s}{2}\right)^2$$

- The second zero does not have a physical meaning in a type 2 compensator

$$f_{z1}=\frac{1}{2\pi\left[\frac{T_{sw}\left(a_0-a_2+\sqrt{a_1^{\,2}-4a_0a_2}\right)}{2\left(a_0+a_1+a_2\right)}\right]}$$

Infinite value

$$f_{z2}=\frac{1}{2\pi\left[-\frac{T_{sw}\left(a_2-a_0+\sqrt{a_1^{\,2}-4a_0a_2}\right)}{2\left(a_0+a_1+a_2\right)}\right]}$$

$$G_0=\frac{a_0+a_1+a_2}{1+b_{11}+b_2}$$

$$b_{11}=b_1+1p$$

$$f_{p1}=\frac{1}{2\pi\left[\frac{T_{sw}\left(\sqrt{b_{11}^{\,2}-4b_2}-b_2+1\right)}{2\left(1+b_{11}+b_2\right)}\right]}$$

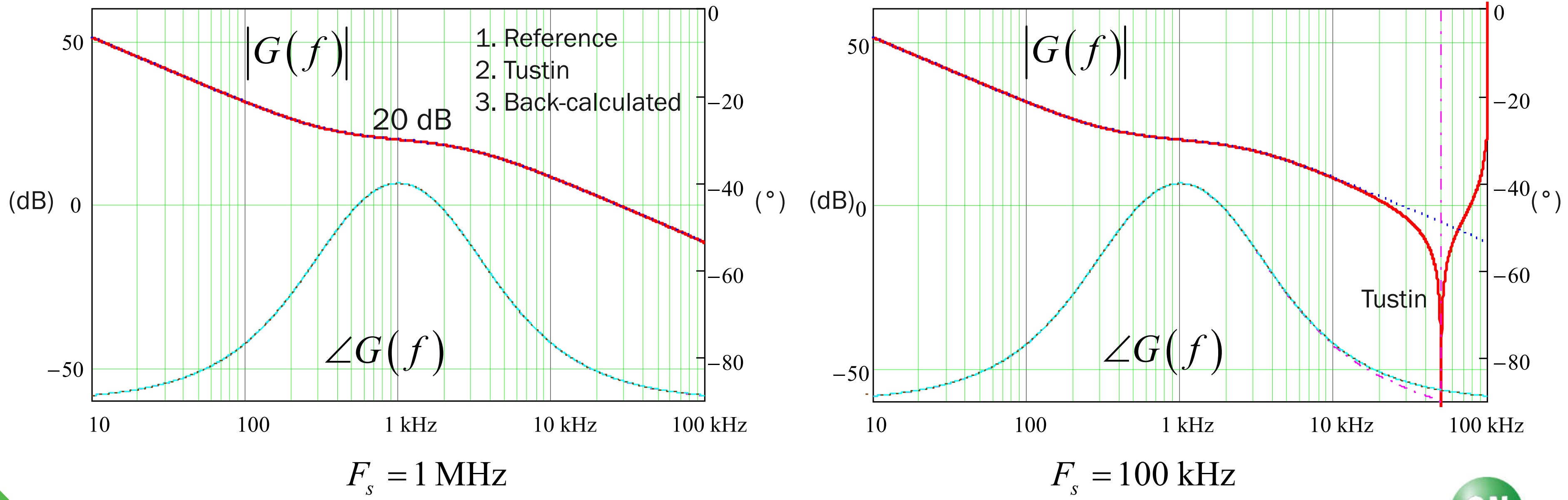
$$f_{p2}=\frac{1}{2\pi\left[-\frac{T_{sw}\left(b_2-1+\sqrt{b_{11}^{\,2}-4b_2}\right)}{2\left(1+b_{11}+b_2\right)}\right]}$$

Testing Coefficients with Mathcad®

- Back-calculate poles and zero values:

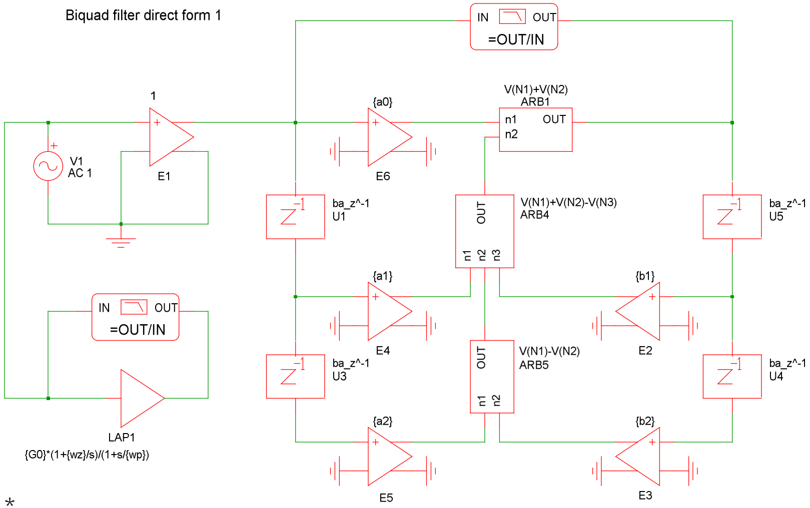
$f_z = 363.9\ \text{Hz}$ $\quad f_{p_1} = 1\ \mu\text{Hz}$ $\quad f_{p_2} = 2.74\ \text{kHz}$ $\quad H_0 \approx 4 \cdot 10^8$ or 172 dB $\quad \dfrac{f_{p_1} G_0}{f_{z_1}} = 20.008\ \text{dB}$ $\quad G(s) = G_0 \dfrac{1 + \dfrac{s}{\omega_z}}{\dfrac{s}{\omega_{p_1}}\left(1 + \dfrac{s}{\omega_{p_2}}\right)}$

Pole at the origin — Infinite OL gain — Mid-band gain



Testing the Filter with SPICE

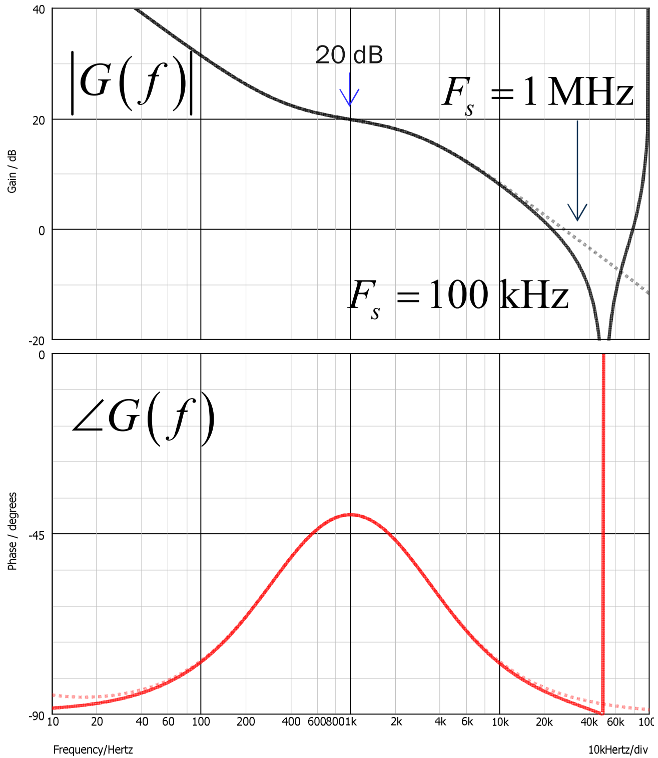
- ❑ Parameters calculation is automated with SIMetrix®




```
*
.PARAM Fsw=100k
.PARAM Tsw={1/Fsw}
.PARAM Ts=Tsw
.PARAM fz=364
.PARAM fp=2.74k
*
.PARAM wz={2*pi*fz}
.PARAM wp={2*pi*fp}
.PARAM G0=10
*
```

20 dB

```
.PARAM a0={G0*Tsw*wp*(Tsw*wz+2)/(2*Tsw*wp+4)}
.PARAM a1={G0*Tsw^2*wp*wz/(Tsw*wp+2)}
.PARAM a2={G0*Tsw*wp*(Tsw*wz-2)/(2*Tsw*wp+4)}
.PARAM b1={-8/(4+2*Tsw*wp)}
.PARAM b2={(4/(Tsw*wp+2))-1}
*
{ '*' } a0 = {a0}
{ '*' } a1 = {a1}
{ '*' } a2 = {a2}
{ '*' } b1 = {b1}
{ '*' } b2 = {b2}
*
```



Implementing a Type 3 Compensator

- ❑ You can now repeat the exercise for a 2-zero/3-pole compensator

$$G(s) = G_0 \frac{\left(1+\frac{\omega_{z_1}}{s}\right)\left(1+\frac{\boxed{s}}{\omega_{z_2}}\right)}{\left(1+\frac{s}{\omega_{p_1}}\right)\left(1+\frac{s}{\omega_{p_2}}\right)} \qquad s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}}$$

$$\left.\begin{aligned} N(z) &= G_0 T_s \omega_{p_1}\omega_{p_2}(z+1)\left(2z+T_s\omega_{z_1}+T_s\omega_{z_1}z-2\right)\left(2z+T_s\omega_{z_2}+T_s\omega_{z_2}z-2\right) \\ D(z) &= 2\omega_{z_2}(z-1)\left(2z+T_s\omega_{p_1}+T_s\omega_{p_1}z-2\right)\left(2z+T_s\omega_{p_2}+T_s\omega_{p_2}z-2\right) \end{aligned}\right\} \text{Divide by } z^3$$

$$G(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}$$

$$a_0 = \frac{G_0 T_s \omega_{p_1}\omega_{p_2}\left(T_s\omega_{z_1}+2\right)\left(T_s\omega_{z_2}+2\right)}{2\left(4\omega_{z_2}+2T_s\omega_{p_1}\omega_{z_2}+2T_s\omega_{p_2}\omega_{z_2}+T_s^2\omega_{p_1}\omega_{p_2}\omega_{z_2}\right)}$$

$$a_1 = \frac{G_0 T_s \omega_{p_1}\omega_{p_2}\left(2T_s\omega_{z_1}+2T_s\omega_{z_2}+3T_s^2\omega_{z_1}\omega_{z_2}-4\right)}{2\left(4\omega_{z_2}+2T_s\omega_{p_1}\omega_{z_2}+2T_s\omega_{p_2}\omega_{z_2}+T_s^2\omega_{p_1}\omega_{p_2}\omega_{z_2}\right)}$$

$$a_2 = -\frac{G_0 T_s \omega_{p_1}\omega_{p_2}\left(2T_s\omega_{z_1}+2T_s\omega_{z_2}-3T_s^2\omega_{z_1}\omega_{z_2}+4\right)}{2\left(4\omega_{z_2}+2T_s\omega_{p_1}\omega_{z_2}+2T_s\omega_{p_2}\omega_{z_2}+T_s^2\omega_{p_1}\omega_{p_2}\omega_{z_2}\right)}$$

$$a_3 = \frac{G_0 T_s \omega_{p_1}\omega_{p_2}\left(T_s\omega_{z_1}-2\right)\left(T_s\omega_{z_2}-2\right)}{2\left(4\omega_{z_2}+2T_s\omega_{p_1}\omega_{z_2}+2T_s\omega_{p_2}\omega_{z_2}+T_s^2\omega_{p_1}\omega_{p_2}\omega_{z_2}\right)}$$

$$b_1 = \frac{T_s\omega_{p_2}-2}{T_s\omega_{p_2}+2} - \frac{4}{T_s\omega_{p_1}+2} \qquad b_2 = \frac{16}{\left(T_s\omega_{p_1}+2\right)\left(T_s\omega_{p_2}+2\right)}$$

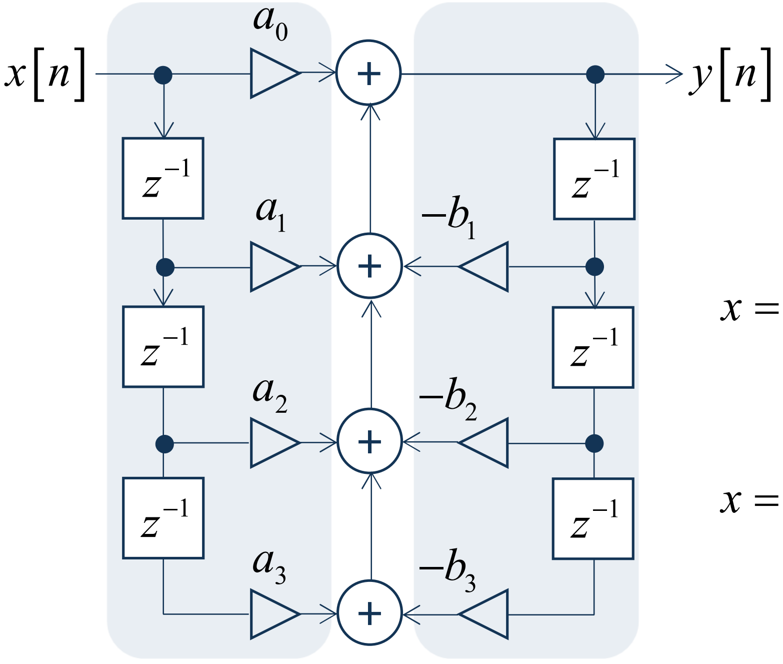
$$b_3 = -\frac{\left(T_s\omega_{p_1}-2\right)\left(T_s\omega_{p_2}-2\right)}{\left(T_s\omega_{p_1}+2\right)\left(T_s\omega_{p_2}+2\right)}$$

Watch for the Computation Flow

- The zeroes are computed first then the poles

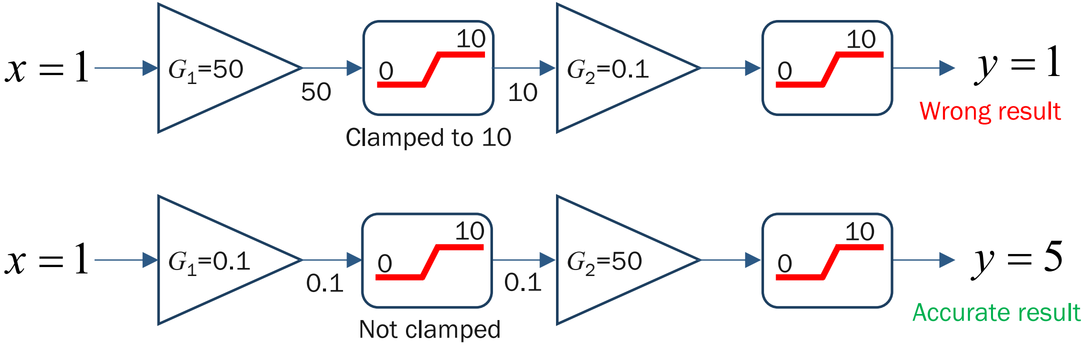
Feedforward coefficients   Feedback coefficients

$$y[n]=a_0x[n]+a_1x[n-1]+a_2x[n-2]+a_3\left[n-3\right]-b_1y[n-1]-b_2y[n-2]-b_3y[n-3]$$



Zeroes calculation   Poles calculation

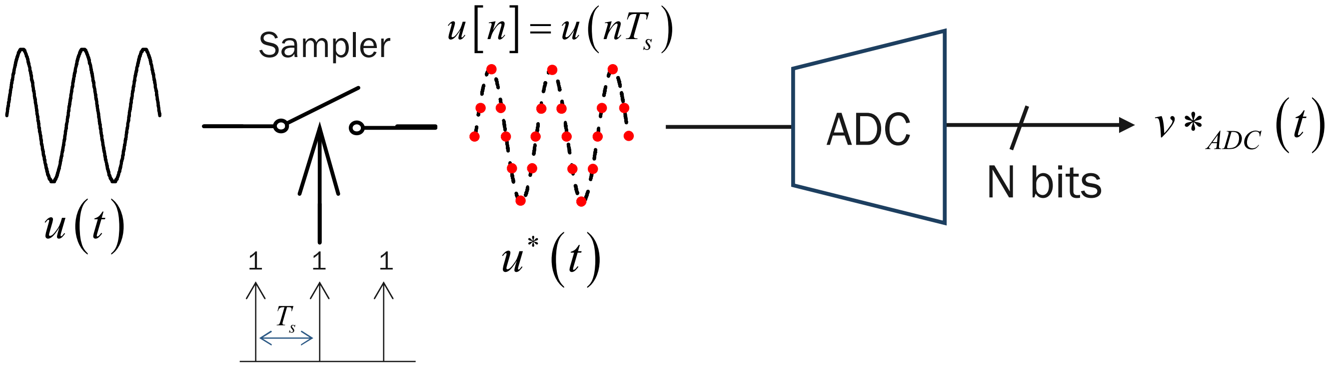
- ✓ Computing zeroes first reduces overflow issues
- ❖ Poles tend to amplify variables when they occur
- ❖ Zeroes attenuate the variable when they occur



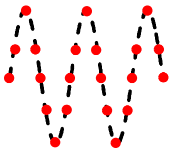
M. Jovanović, *Introduction to Digital Control of Switch-Mode Converters*, in-house course, PHX 2014

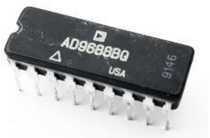
Going Digital – Converting Samples in Words

- The analog-to-digital converter converts the sample in a binary word

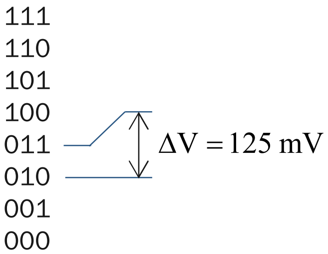


- The conversion process brings a *quantization* error





3-bit ADC



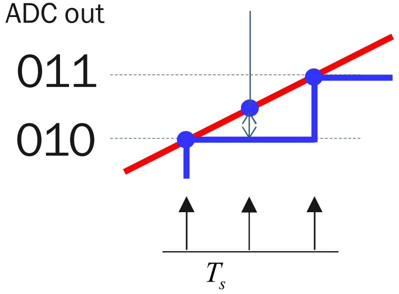
8 quantization levels

Range: 0 to 1 V

Resolution:

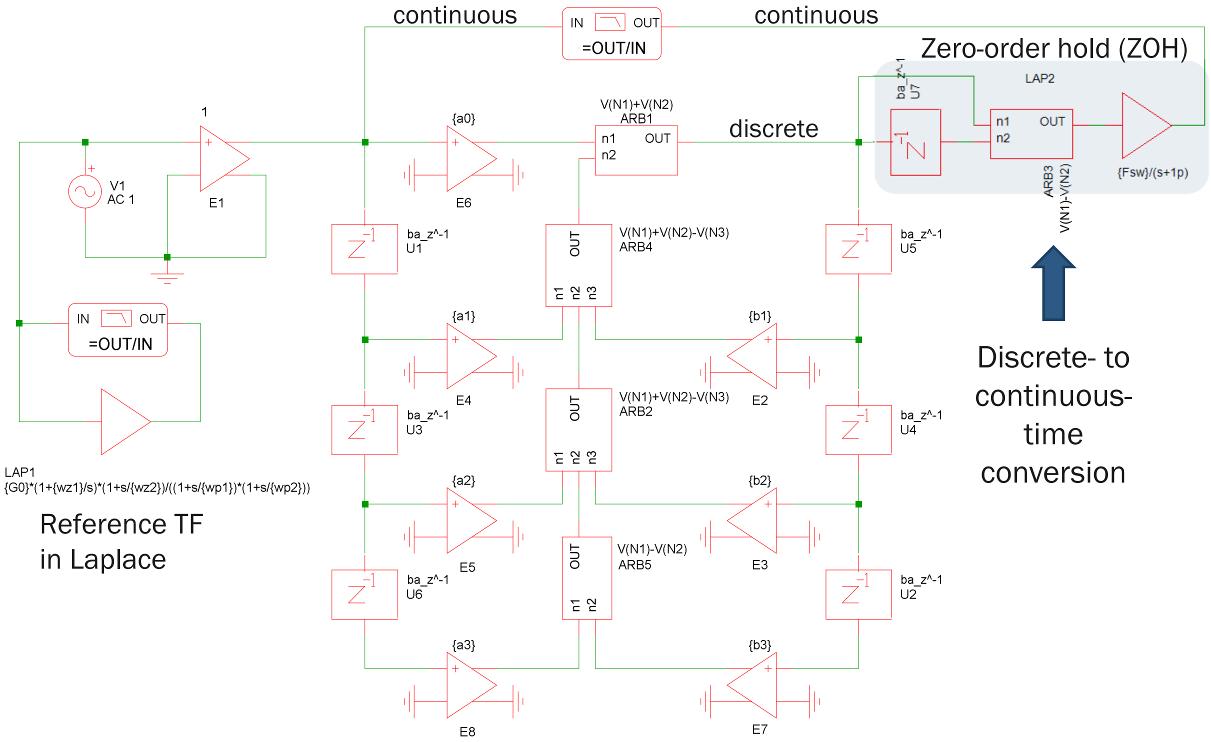
$$\frac{1}{2^3} = \frac{1}{8} = 125\text{ mV}$$

Quantization error



SIMetrix® Lets you Test Coefficient Values

- Before implementation, it is interesting to test the filter in SPICE




```
*
* Enter Design Goals Information Here *
*
.PARAM fc=3k ; targeted crossover
*
.PARAM Fsw=1Meg ; sampling frequency
.PARAM Tsw={1/Fsw} ; sampling period
.PARAM Ts=Tsw
*
*
.PARAM fz1=200
.PARAM fz2=600
.PARAM fp1=21k
.PARAM fp2=21k
.PARAM Gfc=-20 ; magnitude at crossover
*
.PARAM G1={10^(-Gfc/20)} ; needed gain at fc
.PARAM G0={G1/SQRT((1+(fz1/fc)^2)*
+(1+(fc/fz2)^2)/(1+(fc/fp1)^2)*(1+(fc/fp2)^2))}
*
.PARAM wz1={2*pi*fz1}
.PARAM wp1={2*pi*fp1}
.PARAM wz2={2*pi*fz2}
.PARAM wp2={2*pi*fp2}
```

Automated calculation

S. Ben-Yaakov, D. Adar, Generic Average Modeling and Simulation of Discrete Controllers, APEC 2001

SIMPLIS® Offers a Sampled PID Block

- Add the third pole via a RC filter tuned at ω_{p2}

From type 3 conversion

$$G(z) = K_p + \frac{K_i}{A_i(z)} + \frac{K_d A_d(z)}{\gamma K_d A_d(z) + 1}$$

Internal z transfer functions computed with forward- or backward-Euler or Tustin

$$K_p = k_p \qquad K_i = k_i T_s \qquad K_d = \frac{k_d}{T_s}$$

$$\gamma = \frac{1}{\omega_{p_1} k_d}$$

$$0 \le \omega_{p_1,z} \le 1 \rightarrow \gamma k_d \ge \begin{cases} 1 & \text{Forward-Euler} \\ 0 & \text{Backward-Euler} \\ 0.5 & \text{Tustin} \end{cases}$$

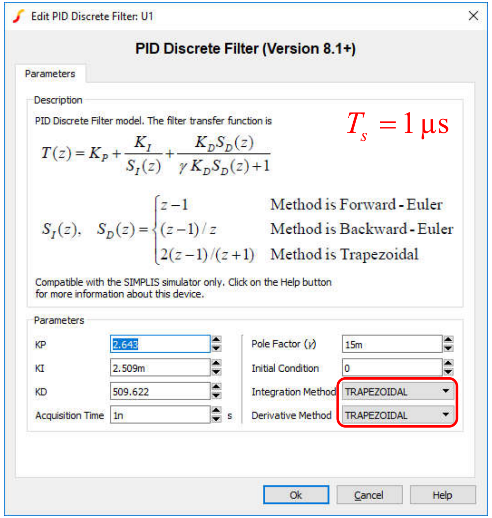
Pole must lie in unity circle

$$\boxed{\begin{aligned} f_c &= 3\text{ kHz} \\ G_{f_c} &= 20\text{ dB} \end{aligned}}$$

$$f_{z_1} = 200\text{ Hz}$$
$$f_{z_2} = 600\text{ Hz}$$
$$f_{p_1} = 21\text{ kHz}$$
$$f_{p_2} = 21\text{ kHz}$$

Externally implemented

$$G_0 = 1.99$$
$$K_p = k_p = 2.643$$
$$K_i = k_i T_s = 2.509m$$
$$K_d = \frac{k_d}{T_s} \approx 510$$
$$\gamma = \frac{1}{\omega_{p_1} k_d} = 15m$$



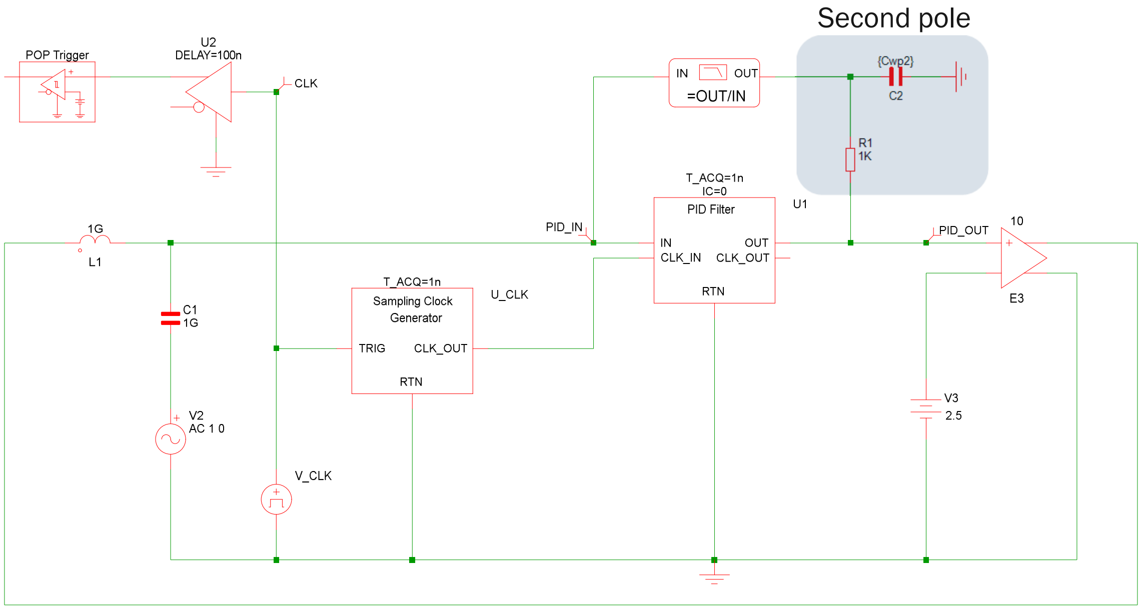
You can choose and combine mapping functions

Public Information

https://www.simplistechnologies.com/documentation/simplis/dp_discrete_filters/topics/piddiscretefilter.htm

The Sampled PID Parameters Can be Automated

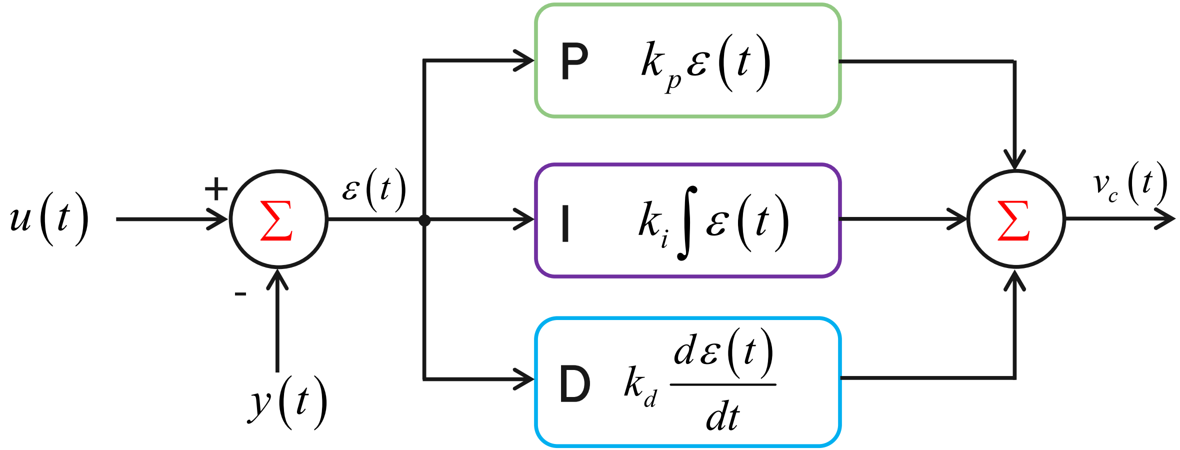
- ❑ Because of infinite quasi-static gain, a biasing circuit is necessary




```
*
.var Fs=1Meg
.var Ts={1/Fs}
.var fc=3k
.var Gfc=10
.var fz1=200
.var fz2=600
.var fp1=21k
.var fp2=21k
*
.var wz1={2*pi*fz1}
.var wz2={2*pi*fz2}
.var wp1={2*pi*fp1}
.var wp2={2*pi*fp2}
*
.var a=sqrt(1+(fc/fp1)^2)
.var b=sqrt(1+(fc/fp2)^2)
.var c=sqrt(1+(fz1/fc)^2)
.var d=sqrt(1+(fc/fz2)^2)
.var G0=(a*b/(c*d))*Gfc
.var Cwp2={1/(wp2*1k)}
*
.var ki=G0*wz1
.var kd=((wz2-wp1)*(ki-G0*wp1))/(wp1^2*wz2)
.var kp=G0*((wz1+wz2)/wz2-wz1/wp1)
*
.var kpz=kp
.var kdz=kd/Ts
.var kiz=ki*Ts
.var gam=1/(wp1*kd)
*
{ '*' }
{ '*' }
{ '*' } G0 = {G0}
{ '*' } ki = {ki}
{ '*' } kd = {kd}
{ '*' } kp = {kp}
{ '*' } kpz = {kpz}
{ '*' } kiz = {kiz}
{ '*' } kdz = {kdz}
{ '*' } gam = {gam}
{ '*' }
{ '*' }
*
```

You Can Also Derive PID Blocks Yourself

- ❑ We can map the continuous-time representation in the *z*-domain



- ➢ The proportional block is straightforward

$$V_c(s)\Big|_{k_i=k_d=0} = k_p\varepsilon(s) \Rightarrow V_c[n] = k_p\varepsilon[n] \Rightarrow \varepsilon[n] \rightarrow \boxed{k_p} \rightarrow k_p\varepsilon[n]$$

Backward-Euler Applied to the Integral Path

- ❑ Map the integral block defined in the s-domain

$$V_c(s)\Big|_{k_p=k_d=0} = \frac{k_p}{\tau_i}\frac{1}{\boxed{s}}\varepsilon(s) \qquad s = \frac{1-z^{-1}}{T_s}$$

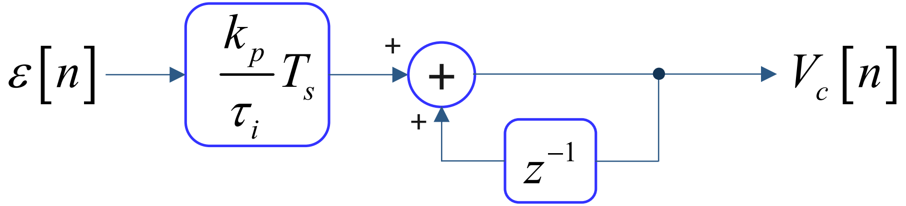
Backward-Euler

$$\Rightarrow \quad V_c(z)\Big|_{k_p=k_d=0} = \frac{k_p}{\tau_i}\frac{1}{\dfrac{1-z^{-1}}{T_s}}\varepsilon(z)$$

- ❑ Develop and rearrange this expression:

$$V_c(z)\left(1-z^{-1}\right) = \frac{k_p T_s}{\tau_i}\varepsilon(z) \quad \Rightarrow \quad V_c[n]-V_c[n-1] = \frac{k_p T_s}{\tau_i}\varepsilon[n]$$

- ➢ Assemble the blocks to form the integral path:



Backward-Euler Applied to the Derivative Path

- Map the derivative block defined in the s-domain

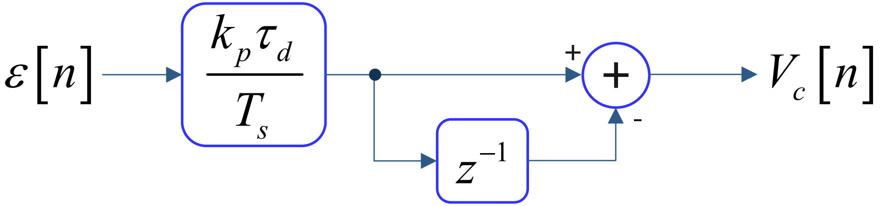
$$V_c(s)\Big|_{k_p=k_i=0} = k_p\tau_d \boxed{s}\,\varepsilon(s) \qquad s = \frac{1-z^{-1}}{T_s} \qquad \Rightarrow \qquad V_c(z)\Big|_{k_p=k_i=0} = k_p\tau_d \frac{1-z^{-1}}{T_s}\varepsilon(z)$$

Backward-Euler

- Develop and rearrange this expression:

$$V_c(z) = \frac{k_p\tau_d}{T_s}\left(1-z^{-1}\right)\varepsilon(z) \qquad \Rightarrow \qquad V_c[n] = \frac{k_p\tau_d}{T_s}\left(\varepsilon[n]-\varepsilon[n-1]\right)$$

- Assemble the blocks to form the derivative path:



A Filter is Needed for the Derivative Term

- Start from the low-pass filter expression

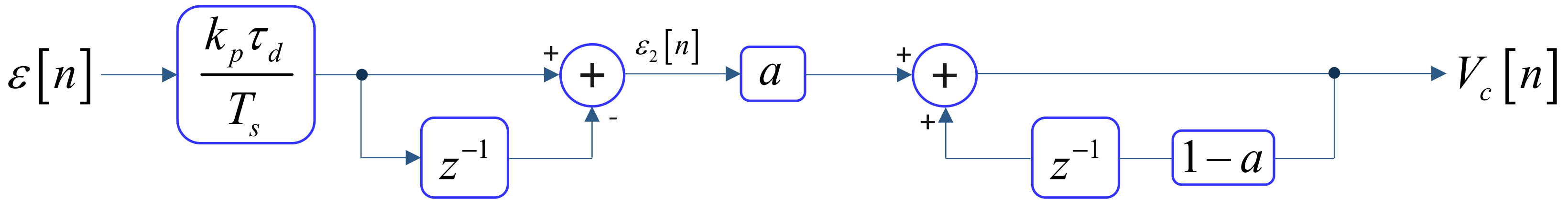
$$G(s)=\frac{1}{1+\frac{s}{\omega_p}} \qquad s=\frac{1-z^{-1}}{T_s} \qquad \omega_p=\frac{N}{\tau_d}$$

$$\Rightarrow \quad G(z)=\frac{1}{\frac{\tau_d}{NT_s}-\frac{\tau_d}{NT_s}z^{-1}+1}=\frac{1}{\left(1+\frac{\tau_d}{NT_s}\right)\left(1-\frac{\tau_d}{\tau_d+NT_s}z^{-1}\right)}$$

- Factor and rearrange this expression considering ε_2 as the input and V_c the output

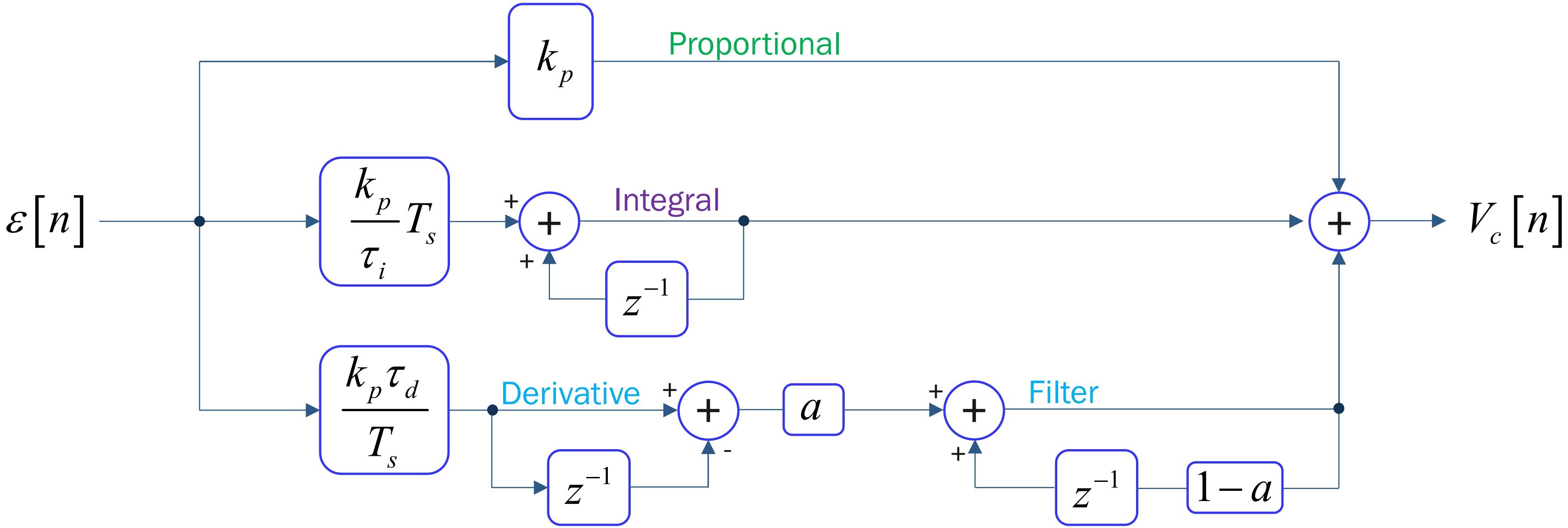
$$G(z)=\frac{a}{1-z^{-1}(1-a)} \qquad a=\frac{T_s}{\frac{\tau_d}{N}+T_s} \quad \Rightarrow \quad V_c[n]=a\cdot\varepsilon_2[n]+V_c[n-1](1-a)$$

- Insert the extra filter block in series with the derivative term



Final Filtered PID Implementation

- ❑ The model can be enriched with an anti-windup structure

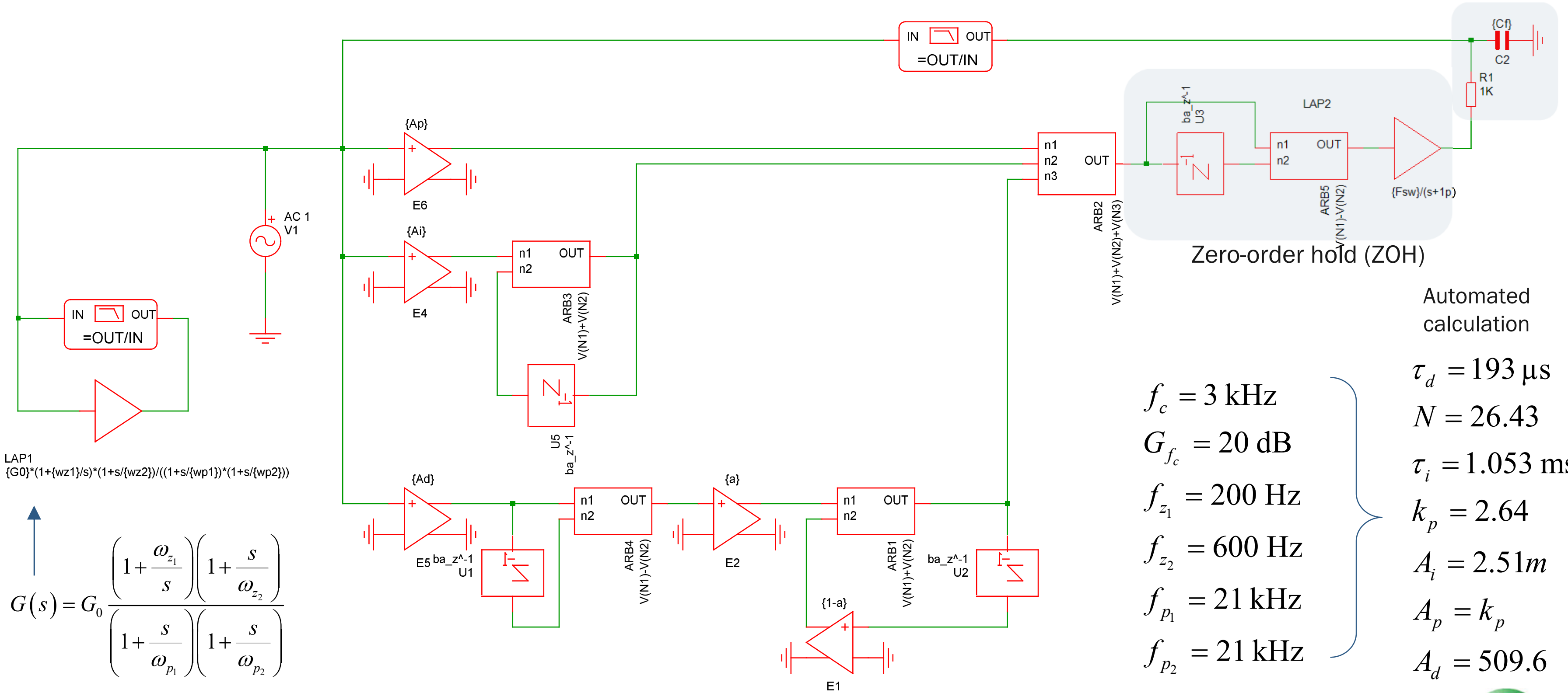


$$\tau_i = \frac{\omega_{z_1} + \omega_{z_2}}{\omega_{z_1}\omega_{z_2}} - \frac{1}{\omega_{p_1}} \quad N = \frac{\omega_{p_1}{}^2}{\omega_{p_1}\omega_{z_1} + \omega_{p_1}\omega_{z_2} - \omega_{z_1}\omega_{z_2}} - 1 \quad k_p = \frac{\omega_{po}}{\omega_{z_1}} - \frac{\omega_{po}}{\omega_{p_1}} + \frac{\omega_{po}}{\omega_{z_2}} \quad \tau_d = \frac{\left(\omega_{p_1} - \omega_{z_1}\right)\left(\omega_{p_1} - \omega_{z_2}\right)}{\left(\omega_{p_1}\omega_{z_1} + \omega_{p_1}\omega_{z_2} - \omega_{z_1}\omega_{z_2}\right)\omega_{p_1}}$$

C. Basso, *Designing Control Loops for Linear and Switching Power Supplies*, Artech House, 2012

Use SIMetrix® to Validate the Structure

- ❑ Simulation lets you verify the blocks are well assembled together



Automated calculation

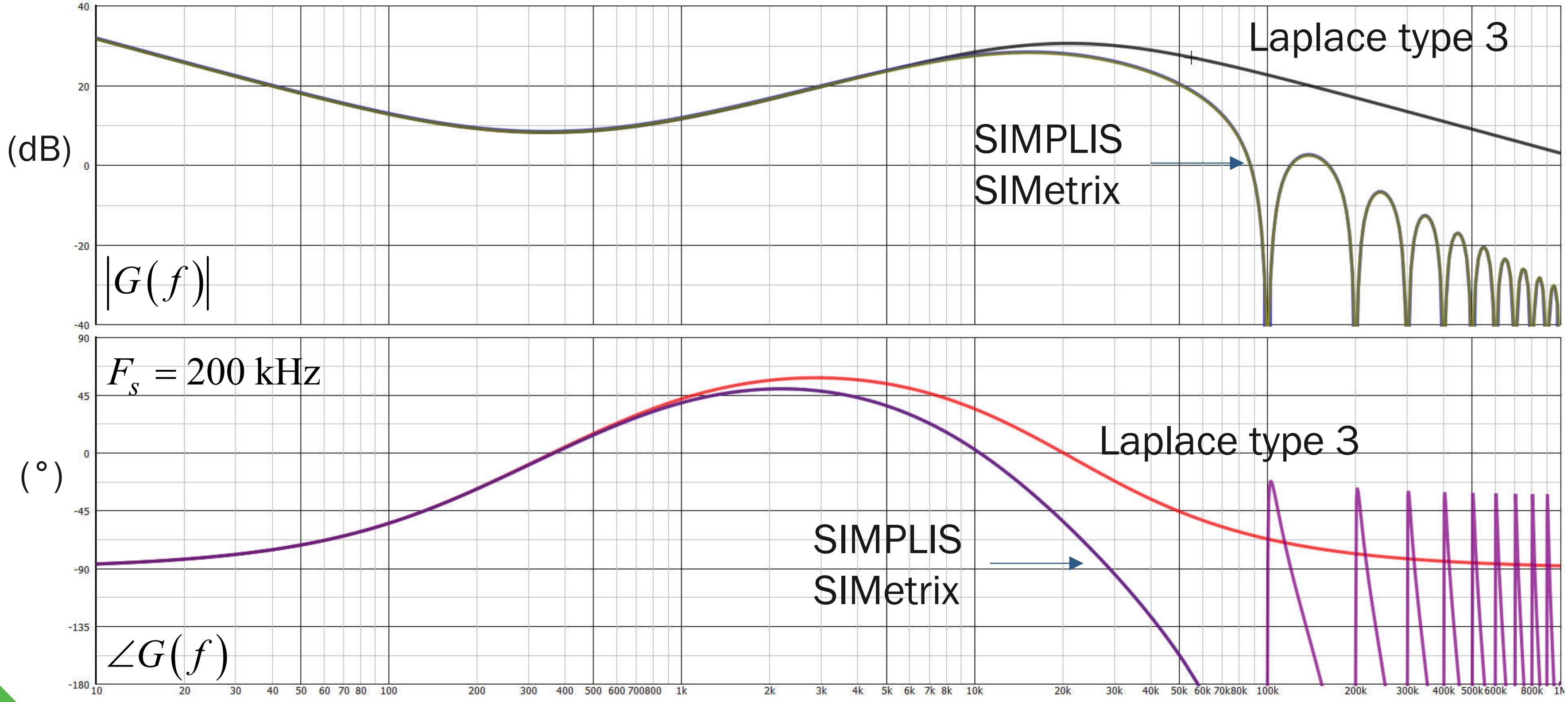
$$G(s) = G_0 \frac{\left(1+\frac{\omega_{z_1}}{s}\right)\left(1+\frac{s}{\omega_{z_2}}\right)}{\left(1+\frac{s}{\omega_{p_1}}\right)\left(1+\frac{s}{\omega_{p_2}}\right)}$$

Reference TF type 3

$$\left.\begin{aligned} f_c &= 3\text{ kHz} \\ G_{f_c} &= 20\text{ dB} \\ f_{z_1} &= 200\text{ Hz} \\ f_{z_2} &= 600\text{ Hz} \\ f_{p_1} &= 21\text{ kHz} \\ f_{p_2} &= 21\text{ kHz} \end{aligned}\right\} \quad \begin{aligned} \tau_d &= 193\ \mu\text{s} \\ N &= 26.43 \\ \tau_i &= 1.053\text{ ms} \\ k_p &= 2.64 \\ A_i &= 2.51m \\ A_p &= k_p \\ A_d &= 509.6 \end{aligned}$$

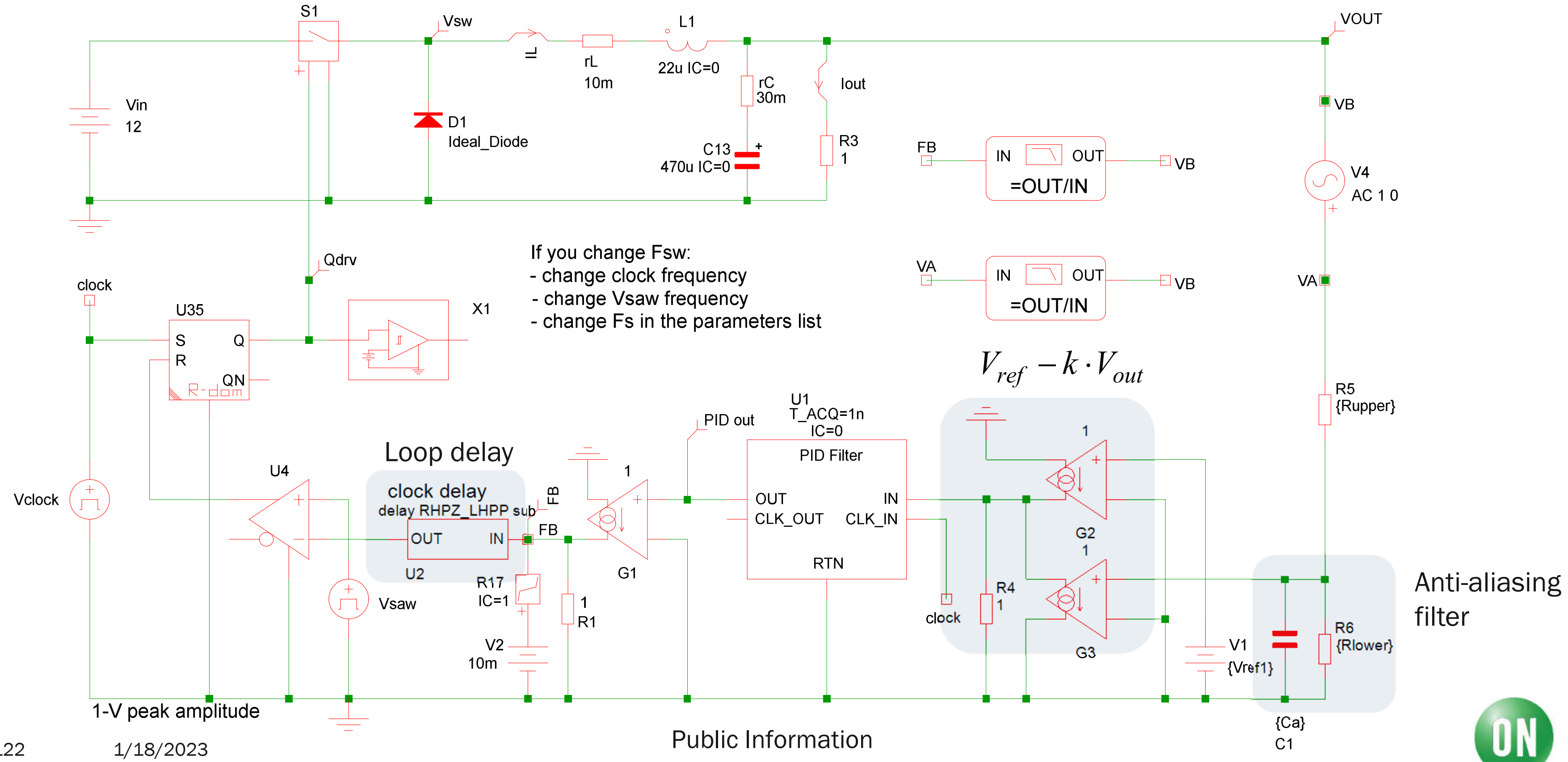
Testing PID Responses

- The SIMPLIS® PID and SIMetrix® discrete version agree well with each others



Practical Application to a Buck Converter

- The PID compensator is applied to a 250-kHz VM buck converter



Automate All Calculations with SIMPLIS®

❑ You can quickly check different compensation strategies and see what it does

```
*
.var Gfc=-8 * magnitude at crossover *
.var PS=-148 * phase lag at crossover *
*
* Enter Design Goals Information Here *
*
.var fc=10k * targetted crossover *
.var PM=60 * choose phase margin at crossover *
*
* Enter the Values for Vout and Bridge Bias Current *
*
.var Vout=5
.var Ibias=250u
.var Vref1=2.5
.var Rlower=Vref1/Ibias
.var Rupper=(Vout-Vref1)/Ibias
.var kdiv=Rlower/(Rlower+Rupper)
*
* Capture the double zero position and one of the pole position *
.var fz1=1.5k
.var fz2=1.5k
.var fp2=50k ; pole is brought by the Tustin mapping at Fsw/2
*
* Do not edit the below lines *
.var boost=PM-PS-90
.var G=(10^(-Gfc/20))/kdiv
.var fp1=fc/tan((2*atan(fc/fz1)-tan(fc/fp2))-boost*pi/180) *
.var Fs=250k
.var Ts={1/Fs}
*
```

Account for the divider network attenuation

```
.globalvar Wtau=2/Ts ; transport delay
.globalvar R={1/(10n*Wtau)}
*
.var Req={Rupper*Rlower/(Rupper+Rlower)} ; equivalent resistance driving Ca
.var Ca={1/(pi*Fs*Req)} ; anti-aliasing filter
*
```

Position the anti-aliasing filter

```
.var wz1={2*pi*fz1}
.var wz2={2*pi*fz2}
.var wp1={2*pi*fp1}
.var wp2={2*pi*fp2}
*
.var a=sqrt(1+(fc/fp1)^2)
.var b=sqrt(1+(fc/fp2)^2)
.var c=sqrt(1+(fz1/fc)^2)
.var d=sqrt(1+(fc/fz2)^2)
.var G0=(a*b/(c*d))*G
.var Cwp2={1/(wp2*1k)}
*
.var ki=G0*wz1
.var kd=((wz2-wp1)*(ki-G0*wp1))/(wp1^2*wz2)
.var kp=G0*((wz1+wz2)/wz2-wz1/wp1)
*
.var kpz=kp
.var kdz=kd/Ts
.var kiz=ki*Ts
.var gam=1/(wp1*kd)
*
```

PID block coefficients

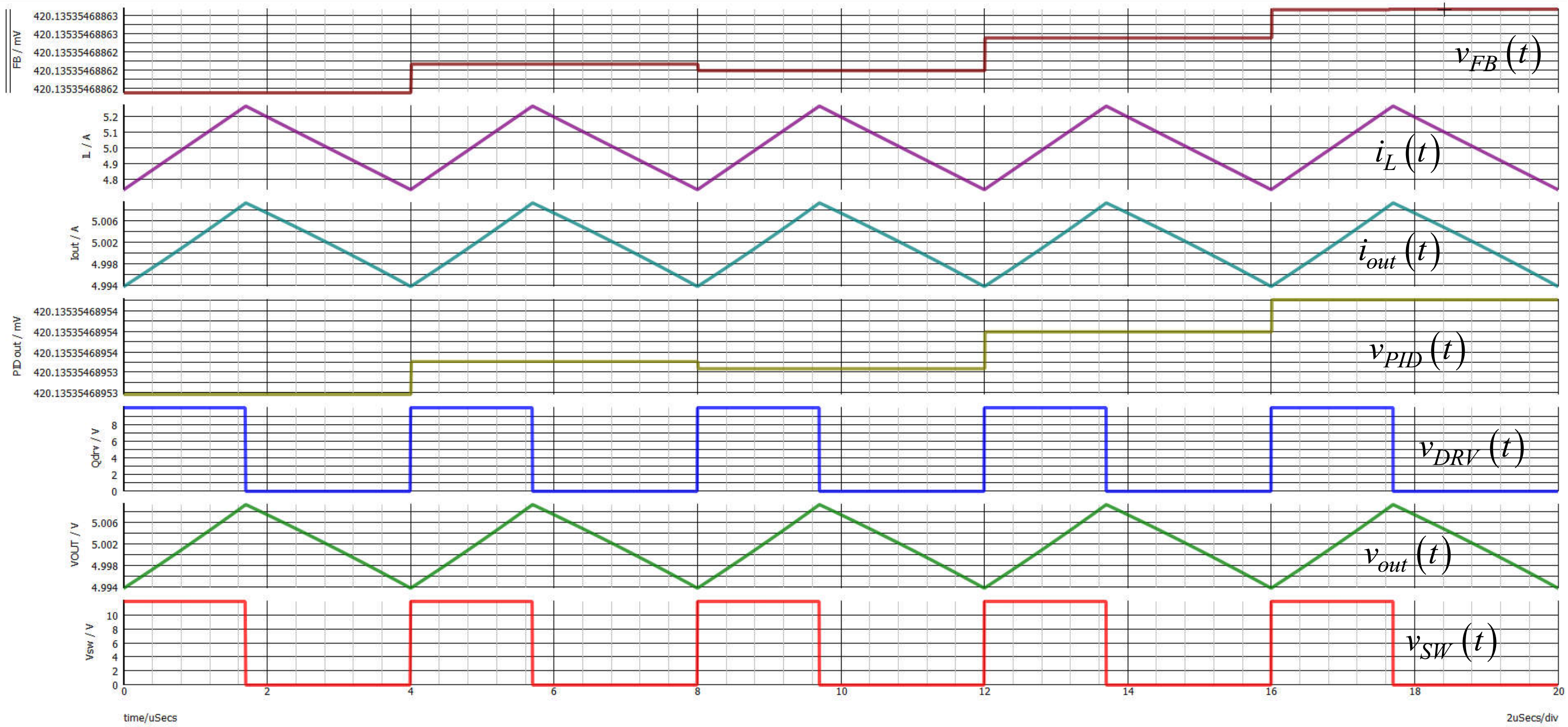
Display coefficients in the netlist

```
*
{ '*' }
{ '*' }
{ '*' } G0 = {G0}
{ '*' } fp1 = {fp1}
{ '*' } ki = {ki}
{ '*' } kd = {kd}
{ '*' } kp = {kp}
{ '*' } kpz = {kpz}
{ '*' } kiz = {kiz}
{ '*' } kdz = {kdz}
{ '*' } gam = {gam}
{ '*' }
{ '*' }
*
```

Public Information

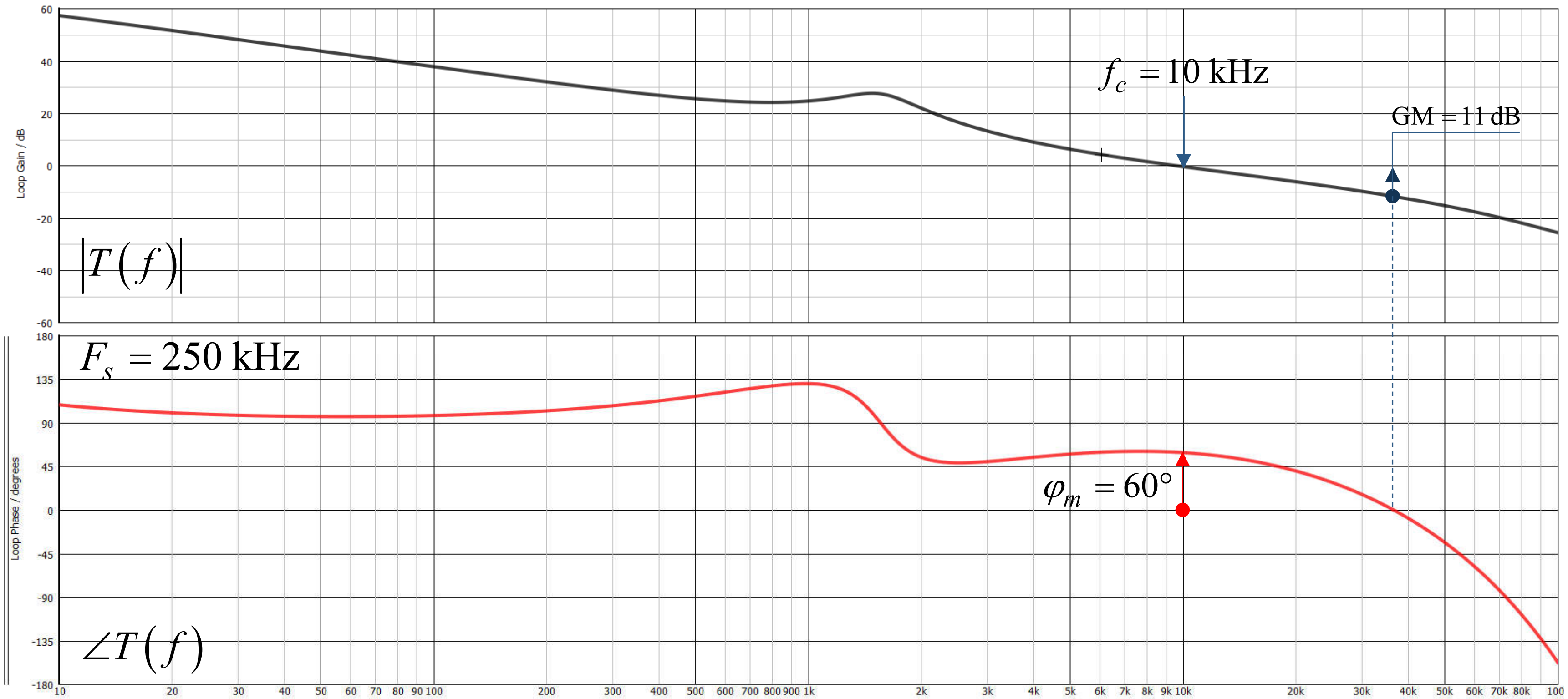
Cycle-by-Cycle Simulation Shows a Stable Operation

- Simulation shows the staircase waveform on the PID output



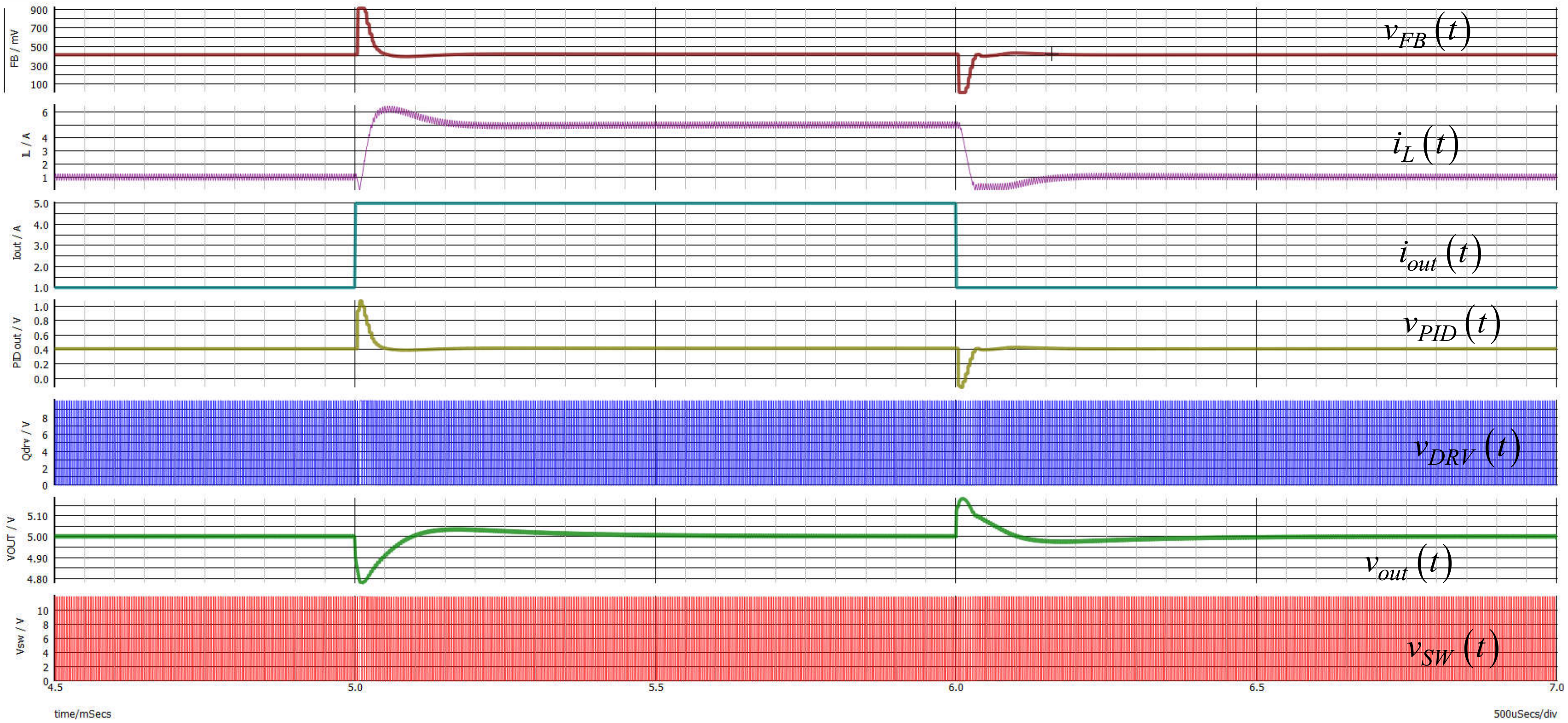
Open-Loop Gain is Obtained in a few Seconds

- ❑ Crossover and phase margin are good at a 250-kHz sampling frequency



Excellent Transient Response

- The recovery is good and fast with a 1-to-5-A transient step (1 µs)



Conclusion

- ❑ To close the loop, you need a compensator
- ❖ The compensator is designed to stabilize and shape the response
- ❖ This filter can be implemented using an op-amp, a TL431, an OTA…
- ❖ An op-amp is not a perfect element and it impacts final results
- ❑ Digital control implements a discrete-time compensator
- ❑ You analyse the stability of the converter in the z-domain
- ❑ Simulation tools are useful to verify architectures before coding
- ❑ Going from the s-domain to the z-domain warps the response
- ❖ Pre-warping poles-zeroes position is the way to go
- ❑ Simulation is an excellent tool for visualizing the effects of discretization
- ❖ Plotting the response before programming is a gain in time!