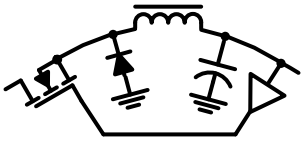


Power Supply Design Seminar

Introduction to Digital Power Control

Texas Instruments

Reproduced from
2026 Texas Instruments Power Supply Design Seminar
SEM2600
Topic 1
Desheng Guo and Sumit Patil
Literature Number: SLUP428

Power Supply Design Seminar resources
are available at:
www.ti.com/psds

This paper covers the fundamentals of microcontroller-based digital power control design. The working principle of building blocks in the digital power control like digital sampling, compensator, and actuators is explained. The essential parameters of each block are discussed along with its effects on control loop performance. A step-by-step design process is shown using the example of voltage-mode controlled buck converter. It also explains feedback circuit design, considering analog to digital converters along with its configurations with respect to the actuator, that is, digital pulse-width modulation. It also explains the selection, implementation, and compensation procedures for the digital compensator.

Introduction

In most systems, power supplies are traditionally controlled using analog methods as shown in **Figure 1**. These solutions are straightforward to implement which only requires the selection of appropriate resistors and capacitors based on controller IC datasheet recommendations. However, designers often encounter limitations with these methods when developing dedicated power products such as charging stations or advanced power conversion systems.

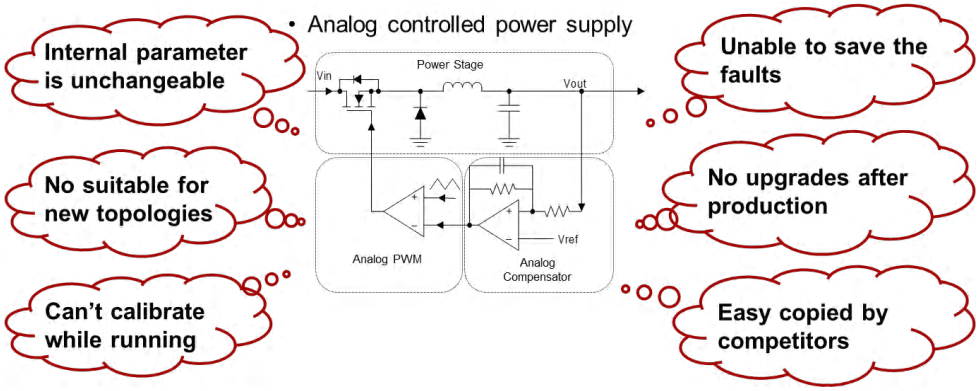


Figure 1. *Basic blocks of simplified analog power control loop.*

One of the examples is updating critical control parameters such as mode-transition thresholds and the compensator coefficients to optimize the loop response. Another challenge arises from the rapid evolution of power topologies such as totem-pole PFC, LLC converters, asymmetrical half-bridge designs, and so forth. In many cases, adapting an analog controller to these advanced topologies is impractical since it requires developing an algorithm that often involves complex computing. Similarly, the applications that demand high accuracy such as those involving batteries with variable impedance benefit from online calibration of parameters, which is difficult to achieve with purely analog control.

Complex use cases present additional requirements, such as the ability to store diagnostic information to aid in troubleshooting equipment failures. These functions generally require a processor with memory, which is not feasible with analog-only solutions. Moreover, post-production firmware updates or parameter updates whether to resolve issues or to optimize the performance are cumbersome with analog controllers, as they often require manual hardware modifications on the PCB. Intellectual property protection is also a concern.

Digital control methods address many of these challenges. Through flexible parameter adjustment, advanced sensing with Analog-to-Digital Converter (ADC), actuation through Digital Pulse Width Modulator (DPWM), and the

implementation of digital compensators, digital control enables greater adaptability, accuracy, and security. This paper presents the essential building blocks of digital power control, using the design of a buck converter as an illustrative example.

Digital Power Control Loop

Digital control of power converters can be understood through three essential steps: Sample, Compensate, and Actuate. A simple digitally controlled buck converter, as illustrated in **Figure 2**, demonstrates these building blocks.

In the first step, the analog feedback signal is converted into digital values using an ADC. This process samples and quantizes the continuous feedback signal into discrete numerical values. The discretization in time and amplitude marks a fundamental distinction between digital and analog control approaches. In the second step, the digitized feedback is compared with a reference value to determine the error. A digital compensator then processes this error in discrete time to generate the appropriate control effort, which drives the system toward the desired operating point. Because the feedback is sampled, all compensation must also operate in discrete time. The third step involves actuation. The digital controller adjusts the actuator to generate the appropriate switching signals for the power stage. The actuator is often the most complex element of the design, as it must be tailored to the specific topology and control method. To ensure deterministic and precise responses, the actuator is implemented in hardware logic within the MCU. The DPWM module is the central component of this actuator stage. In cases such as current-mode control, the DPWM must also coordinate additional peripherals, including digital ramp generators, Digital-to-Analog Converters (DAC), and comparators to achieve stable and accurate operation.

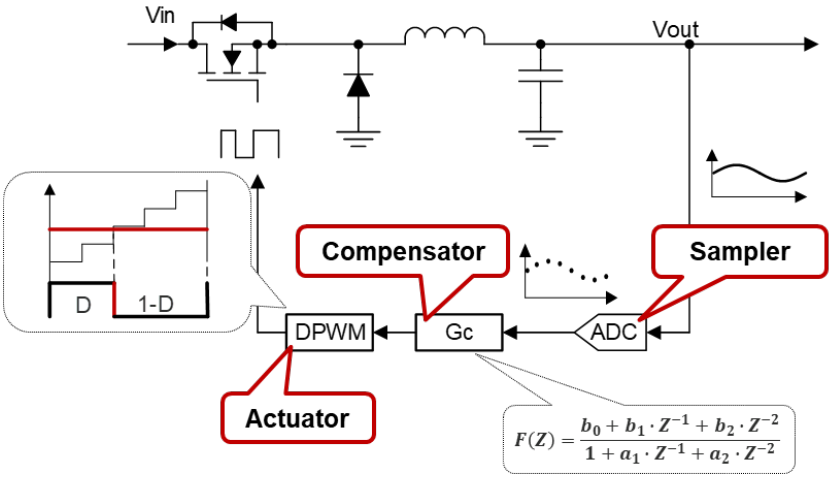


Figure 2. *Basic blocks of simplified digital power control loop.*

Sensing: Analog-to-Digital Converter (ADC)

Digital control depends on an ADC to transform all analog feedback inputs into digital values required for effective control. Most real-time microcontrollers utilize the Successive Approximation Register (SAR) ADC because it offers an excellent balance between speed, resolution, and power efficiency.

While the detailed operation of SAR-ADCs may not be necessary to know, understanding some essential concepts is important for proper use. First, the ADC does not operate continuously but is triggered by a Start of Conversion (SOC) signal. This means feedback values are captured only at specific instants, and all information between SOC triggers is

lost. This is a key distinction between digital and analog control. The rate at which SOC triggers occur is called the sampling frequency, and it determines the frequency range that digital control can influence. Second, the ADC output is not continuous but quantized into discrete levels defined by a limited number of bits. The smallest resolution step is known as the Least Significant Bit (LSB), which introduces what is called quantization error. Third, the conversion process is not completed in a single step but carried out bit by bit, as illustrated in **Figure 3**. This conversion time introduces a delay in the control loop, which reduces the effective phase margin of the system.

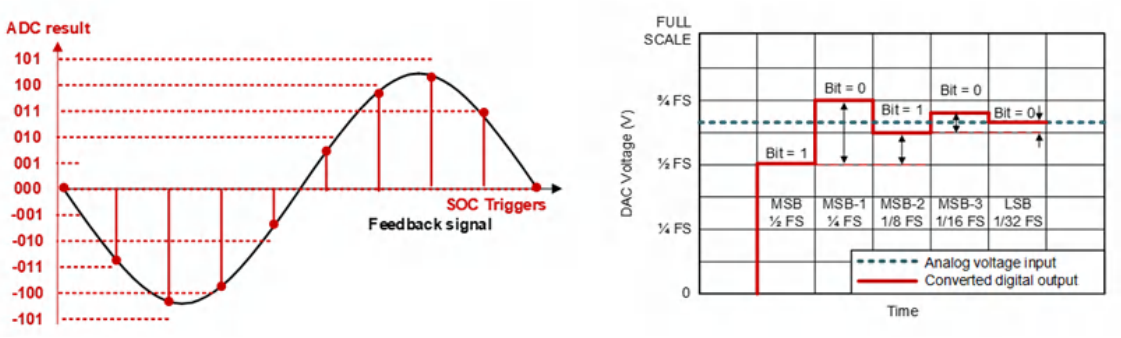


Figure 3. *Sampling and conversion process.*

Acquisition and Conversion

The acquisition process of SAR-ADC needs to sample the external signal using its internal sample and hold (S/H) capacitor. This process not only needs time to settle down, as shown in **Figure 4**, but also will draw a small portion of charge and disturb the input signal. Therefore, it is strongly recommended to put a buffer in front of the ADC input. This could avoid the sampling disturbance and reduce the settling time of the acquisition process. Reduction in settling time (t_{AQ}) is crucial to reduce overall control loop latency. If an external RC filter time constant, $t_{RC} > t_{AQ}$, then inaccurate measurement may occur. But too short t_{RC} may result in not filtering unwanted high frequency at all. So, the correct guideline is $t_{RC} < 5 \times t_{AQ}$. After the acquisition, there is a fixed ADC conversion time t_{CONV} required to complete the analog-to-digital conversion process which is specified by the device datasheet.

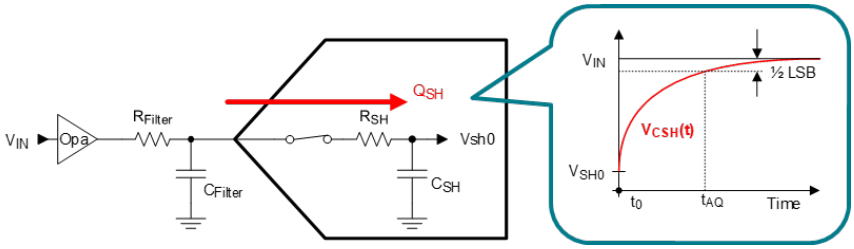


Figure 4. *Acquisition process.*

Sampling and Antialiasing

Since an ADC functions by sampling signals, it is essential to apply the Nyquist sampling criterion. This rule requires that the sampling frequency be at least twice the highest frequency component of the input signal. If this condition is not satisfied, aliasing will occur, as illustrated in **Figure 5**.

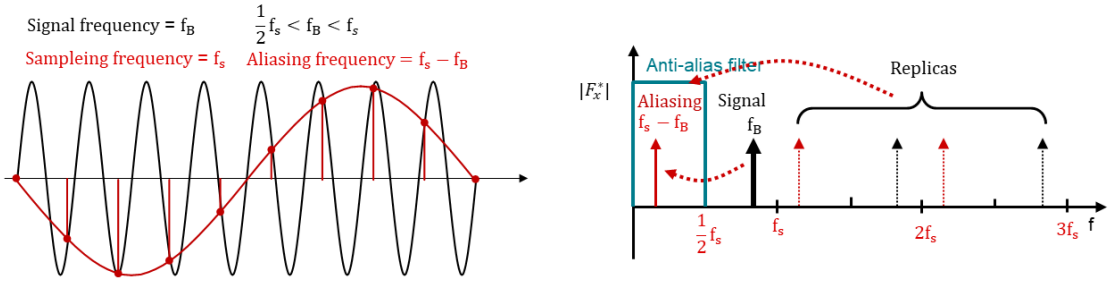


Figure 5. *Sampling and antialiasing.*

For example, if the feedback signal frequency (Fb) is lower than the sampling frequency (Fs) but greater than half of Fs (Fs/2), it will fold into the lower frequency range. Likewise, any frequency component higher than Fs will be mirrored into the sampled band, making all information above Fs/2 indistinguishable. This phenomenon is especially critical in digital control systems. When high-frequency noise is aliased into the sampled band, it appears as a low-frequency signal and becomes embedded in the digital data, making it extremely difficult to eliminate. Therefore, in digital control design, the sampling frequency must be at least twice the bandwidth of interest, and an anti-aliasing filter should be used to block noise beyond the Nyquist frequency.

Quantization Error

A common misconception is that digital control is free from noise, but this is not the case. Because the controller represents signals with a limited number of bits, the values are not continuous but instead change in discrete steps. As a result, each conversion is rounded to the nearest available value, introducing a *rounding error*. As illustrated in **Figure 6**, this error appears as a triangular waveform in the time domain and as a white noise floor in the frequency domain. The noise level is directly tied to the effective number of bits (ENOB) that ADC can provide. Higher bit depth provides finer steps and therefore reduces quantization noise. This error directly affects the control loop performance in terms of the controller's response to small signal errors in feedback which may result in limit cycle oscillations and harmonic distortion in power stage.

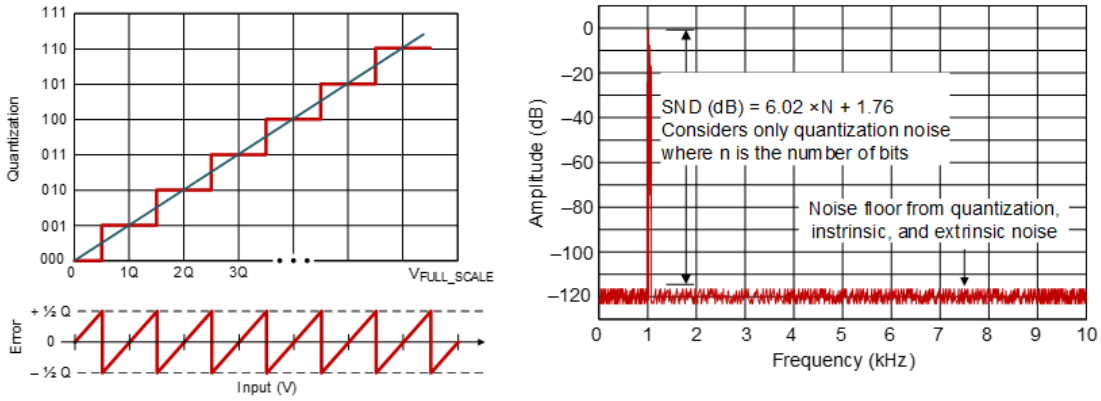


Figure 6. *Quantization error in time and frequency domain.*

Actuator: Digital Pulse Width Modulation (DPWM)

DPWM Working Principle

The Digital Pulse Width Modulator (DPWM) functions like analog PWM but offers enhanced flexibility that allows designers to generate more complex waveforms. Unlike analog implementations, the DPWM operates in discrete time,

comparing a digital counter with a digital comparator to determine switching events. The duty cycle is controlled by adjusting the digital compare value (CMP), while the switching frequency is set by modifying the counter period (PRD). As shown in **Figure 7**, the counter can be configured in UP-count or DOWN-count mode to generate edge-aligned PWM, or in UP-DOWN mode to produce center-aligned PWM. While edge-aligned PWM is sufficient for many power topologies, center-aligned PWM is essential in advanced techniques such as space vector modulation for AC-DC converters, and is also valuable in DC-DC converters like LLC and PSFB, where maintaining a 50% duty cycle is critical.

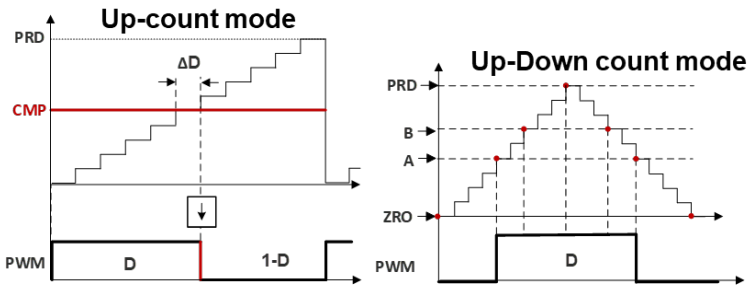


Figure 7. *Counter mode for edge aligned vs center aligned PWM.*

Modern DPWM peripherals extend functionality further by supporting multiple compare events within a single switching cycle, enabling more complex waveform shaping essential for topologies like a Matrix Converter. They can also generate internal and external triggers to synchronize other system operations, such as ADC sampling. These triggers are particularly useful in advanced control methods like peak current mode control, where precise coordination is required between digital ramp amplitude, ramp slope, and DAC.

DPWM Resolution

Since DPWM operates based on a clock counter, it cannot be adjusted in the continuous domain. Its minimum step is one DPWM clock, which raises the issue of finite resolution. DPWM resolution is not related to the number of bits in the counter or comparator. It only depends on the ratio of the PWM clock frequency and the switching frequency. DPWM resolution in percentage can be determined by the ratio of switching frequency to DPWM clock frequency, represented by **Equation 1**. PWM resolution will define the control resolution, which is the smallest step with the minimum control adjustment.

$$\Delta D = \frac{T_{pwmclk}}{T_s} = \frac{f_s}{f_{pwmclk}} \quad (1)$$

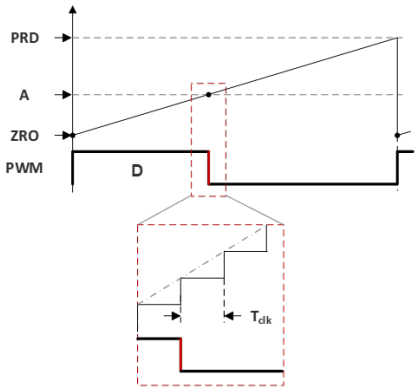


Figure 8. *Discrete counter and the minimum adjustment.*

Limit Cycle Oscillation

Limit cycle oscillation occurs when the control resolution of the DPWM is coarser than the sensing resolution of the ADC. As shown in **Figure 9**, the sensed output voltage cannot precisely match the control target for each DPWM duty value. Instead, the duty cycle continually toggles between the two nearest levels, creating a repeating oscillation. This phenomenon typically results in small-amplitude, high-frequency oscillation that may be acceptable in low-bandwidth systems. However, in high-bandwidth applications such as server or data center power supplies these oscillations can become problematic, leading to increased output ripple and reduced overall performance.

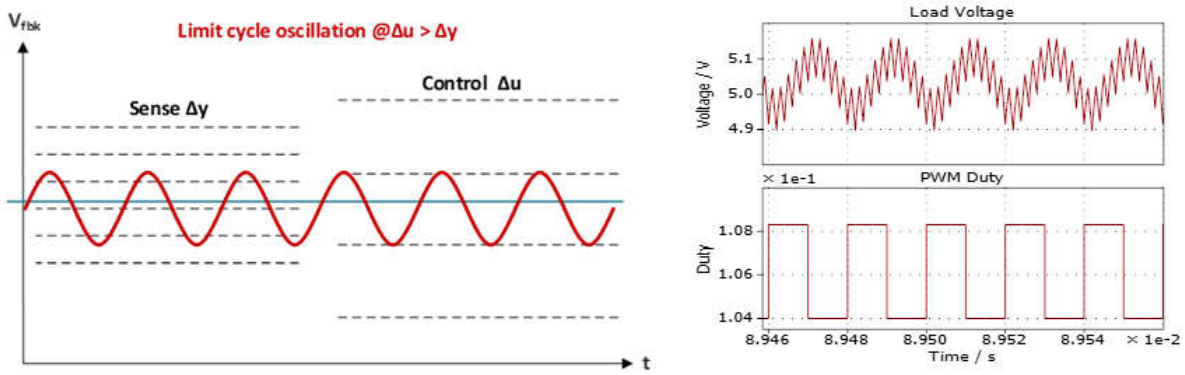


Figure 9. *Limit cycle oscillation and the effect of output voltage.*

The following example showcases the effect of DPWM on output voltage ripple. It shows a 48V to 5V buck converter with 500kHz switching frequency. The calculations in **Table 1** show limit cycle oscillation amplitude using DPWM in normal mode. In this mode, the voltage ripple is 0.2V, which is around 4% of the output voltage. Most of the advanced MCU peripherals have a high-resolution (HR) mode to address this problem. For example, HRPWM mode in F28P55x MCU in C2000™ series MCU can achieve a PWM step size as low as 75ps, which can suppress the oscillation down to 0.036%. That is almost 111 × improvement. **Table 1** compares DPWM in normal mode or HR mode to show the effect of PWM resolution on the output voltage ripple.

Parameter	Normal Mode	HRPWM Mode
Duty adjustment step size	$\Delta D = f_{sw}/f_{pwmclk}$	75ps for F28P55x
ΔV_{PP}	$V_{in} \times \Delta D = 0.2V$	$V_{in} \times t_{step}/t_{pwm} = 1.8mV$
$\%\Delta V_{PP}$	4%	0.036%

Table 1. *Effect of DPWM step size on output voltage ripple.*

Compensator: Digital Compensator

Implementation

Similar to the analog compensator, the digital compensator is used to generate the adjusting control effort value to ensure the control loop is stable and fast enough. However, the input of the digital compensator is the sampled sequence. The digital compensator is described in the discrete sequence. The compensator is executed in discrete time, digitized quantities, and the system is limited by the Nyquist frequency. The differential equation of the digital compensator is also discrete, see **Equation 2**, which is derived from its analog transfer function by using trapezoidal approximation. The digital compensator is presented by the control law diagram, shown in **Figure 10**.

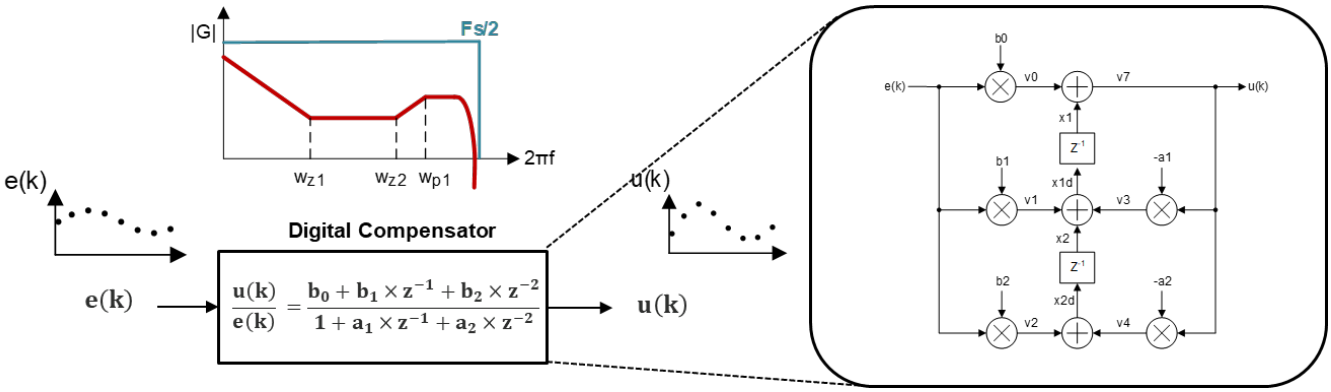


Figure 10. *Digital compensator and implementation.*

$$G(s) = \frac{K_{DC}}{s} \times \frac{\left(1 + \frac{s}{\omega z1}\right)\left(1 + \frac{s}{\omega Z2}\right)}{\left(1 + \frac{s}{\omega_{p1}}\right)} \tag{2}$$

Using trapezoidal approximation by substituting 's' with **Equation 3**:

$$\frac{u(k)}{e(k)} = \frac{b_0 + b_1 \times z^{-1} + b_2 \times z^{-2}}{1 + a_1 \times z^{-1} + a_2 \times z^{-2}} \tag{3}$$

The following section shows the C-code snippet for implementing a second-order direct form 2 controller.

```
// Defines the DCL_DF22 controller structure
typedef
            struct dcl_df22
{
    float32_t b0;   //!< b0
    float32_t b1;   //!< b1
    float32_t b2;   //!< b2
    float32_t a1;   //!< a1
    float32_t a2;   //!< a2
    float32_t x1;   //!< x1
    float32_t x2;   //!< x2
    DCL_DF22_SPS *sps;  //!< Pointer to the shadow parameter set
    DCL_CSS *css;   //!< Pointer to the common support structure
} DCL_DF22;
// Executes a full 2nd order Direct Form 2 controller
static
            inline float32_t DCL_runDF22_C4(DCL_DF22 *p, float32_t ek)
{
    float32_t v7;
    v7 = (ek * p->b0) + p->x1;
    p->x1 = (ek * p->b1) + p->x2 - (v7 * p->a1);
    p->x2 = (ek * p->b2) - (v7 * p->a2);
    return(v7);
}
```

Types of Compensators

Among compensators, the Proportional-Integral (PI) controller is the most widely used. **Table 2** illustrates both the analog and digital implementations. In the analog version, a resistor and capacitor form one pole and one zero. Conversely, the digital PI controller is expressed as a set of discrete equations, but the Bode plot closely matches that of the analog implementation. In analog systems, the PI controller is represented by a continuous-time differential equation, which can be analyzed using the Laplace transform in the s-domain. In digital systems, however, the controller is described in discrete time and analyzed in the z-domain using the z-transform. One key advantage of digital PI control is the ability to directly tune the proportional gain (Kp) and integrator gain (Ki), which can be configured to approximate the behavior of an analog PI controller. When converting from the analog to the digital domain, the backward Euler approximation is commonly applied to map the coefficients. While this method provides a practical conversion, it is not the exact stable operating region of the digital PI controller is actually expanded compared to the analog version. This means that if the analog system is stable, its digital counterpart will also remain stable, but the reverse is not necessarily true.

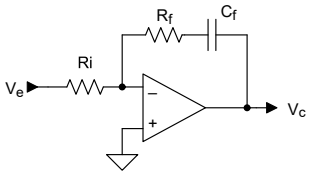
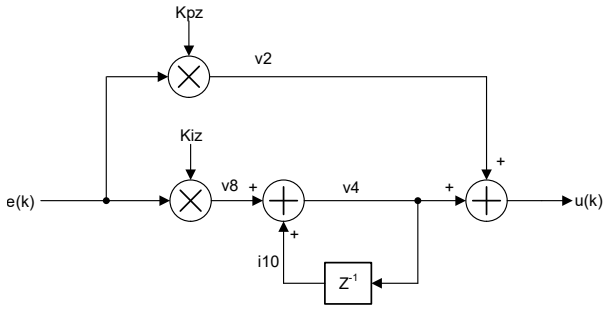
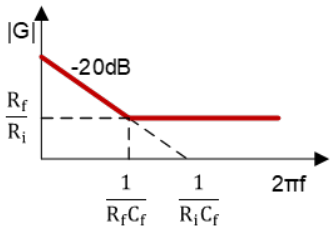
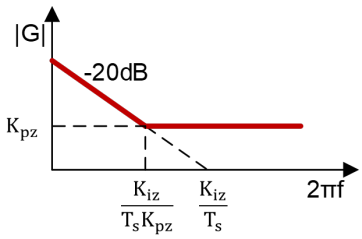
Domain	Analog	Digital
Implementation		
Response		
Time Domain	$\frac{dV_c}{dt} = -\frac{R_f}{R_I} \times \frac{dV_e}{dt} - \frac{1}{R_I \times C_f} \times V_e$	$u(k) = e(k) \times K_{pz} + v(k)$ $v(k) = v(k+1) + e(k) \times K_{iz}$
Frequency Domain	$V_c(s) = -V_e(s) \times \left(K_{Ps} + K_{Is} \times \frac{1}{s}\right)$	$u(z) = e(z) \times \left(K_{pz} + K_{iz} \times \frac{1}{1 - Z^{-1}}\right)$
Transfer Function	$G(s) = K_{Ps} + K_{Is} \times \frac{1}{s}$	$G(z) = K_{pz} + K_{iz} \times \frac{1}{1 - Z^{-1}}$
Coefficients	$K_{Ps} = \frac{R_f}{R_i}; K_{Is} = \frac{1}{R_i C_f}$	$K_{pz} = K_{Ps}$ $K_{iz} = K_{Is} \times T_s$

Table 2. *Analog PI compensator coefficients and the backward Euler approximation.*

Another widely used compensator is the 2P2Z, often referred to as a Type 2 compensator in analog systems. **Table 3** compares the analog and digital implementations. In the analog version, resistors and capacitors form one pole-zero pair, an additional zero, and a non-origin pole. The corresponding digital compensator replicates the same arrangement of poles and zeros, but the implementation is carried out numerically and represented through a control block diagram. In digital form, constructing the two poles of the Type 2 compensator requires a second-order difference equation. To translate the analog transfer function into digital form, techniques such as the trapezoidal (Tustin) approximation are commonly applied. This allows the A and B coefficients of the digital controller to be derived from the analog transfer function. It is important to note; however, that this approximation is not exact. The conversion remains accurate only when the operating frequency is much lower than the Nyquist frequency (half the sampling frequency). Because the transformation is not straightforward, engineers typically rely on software tools such as MATLAB® or Mathcad® to perform the coefficient conversion and verify accuracy.

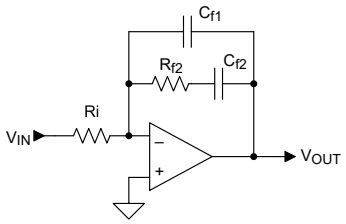
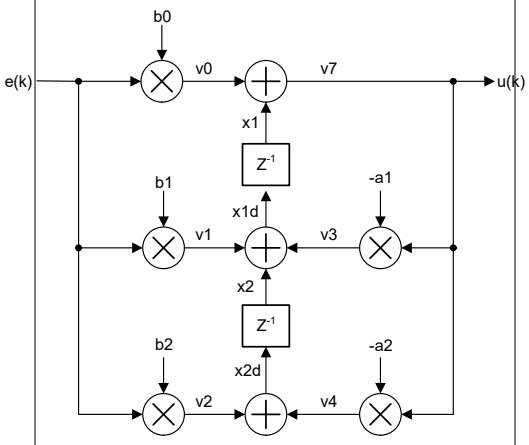
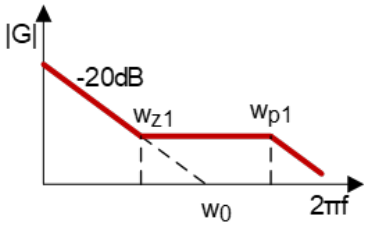
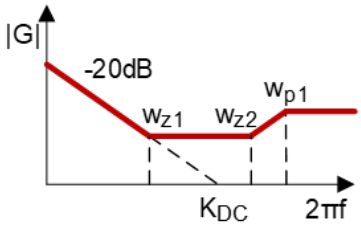
Domain	Analog	Digital
Implementation		
Response		
Transfer Function	$G(s) = \frac{\omega_0}{s} \times \frac{\left(1 + \frac{s}{\omega z1}\right)}{\left(1 + \frac{s}{\omega p1}\right)}$	$G\left(z\right) = \frac{u(k)}{e(k)} = \frac{b_0 + b_1 \times z^{-1} + b_2 \times z^{-2}}{1 + a_1 \times z^{-1} + a_2 \times z^{-2}}$
Coefficients	$\omega_0 = \frac{1}{R_i \times C_{f1} + C_{f2}}; \omega_{z1} = \frac{1}{R_{f2} \times C_{f2}}$ $\omega_{p1} = \frac{1}{\frac{C_{f1} \times C_{f2}}{C_{f1} + C_{f2}} \times R_{f2}}$	$M = \frac{K_{DC} \times \frac{\omega_{p1}}{\omega_{z1} \times \omega_{z2}}}{\frac{2}{T_s} \times \left(\omega_{p1} + \frac{2}{T_s}\right)}$ $a_1 = \frac{-\frac{4}{T_s}}{\omega_{p1} + \frac{2}{T_s}}, a_2 = \frac{\omega_{p1} - \frac{2}{T_s}}{\omega_{p1} + \frac{2}{T_s}};$ $b_0 = M \times \left(\omega_{z1} + \frac{2}{T_s}\right) \times \left(\omega_{z2} + \frac{2}{T_s}\right);$ $b_1 = M \times \left[\left(\omega_{z1} + \frac{2}{T_s}\right) \times \left(\omega_{z2} - \frac{2}{T_s}\right) + \left(\omega_{z1} - \frac{2}{T_s}\right) \times \left(\omega_{z2} + \frac{2}{T_s}\right)\right];$ $b_2 = M \times \left(\omega_{z1} - \frac{2}{T_s}\right) \times \left(\omega_{z2} - \frac{2}{T_s}\right)$

Table 3. *Analog Type 2 compensator coefficients and the direct 2 form using trapezoidal approximation.*

Anti-windup for Integrator

In analog control systems, anti-windup is part of an analog compensator which is an op-amp based feedback network and the integrator gets clamped by the op-amp supply and feedback component impedances. In contrast, digital control systems have numerical based digital integrators in the compensator which has a wider range to accumulate error. So, careful attention must be given to integrator saturation and the use of anti-windup techniques. As illustrated in **Figure 11**, the behavior of a compensator with and without anti-windup can differ significantly. Because the actuator input range is much narrower than the numerical range of the controller, the integrator can continue accumulating errors even after

the compensator output has reached the saturation limit of the actuator. When this happens, the error persists, and the integrator value grows far beyond the actuator's limits. Once the disturbance subsides, the integrator requires a long time to settle back into its valid operating range. During this period, the controller is unable to respond effectively to new changes, leading to sluggish system performance. As shown in **Figure 12**, the recovery of the control signal u(t) without anti-windup is slow compared to the actuator saturation range sat(u). The anti-windup mechanism prevents this problem by detecting actuator saturation and halting further integrator accumulation. This prevents deep saturation and allows the compensator to recover much more quickly, significantly improving transient response. The effect of anti-windup is especially valuable when handling large disturbances. Therefore, if a PI controller exhibits a slow recovery after a transient event, one of the first aspects to check is whether anti-windup protection has been properly implemented.

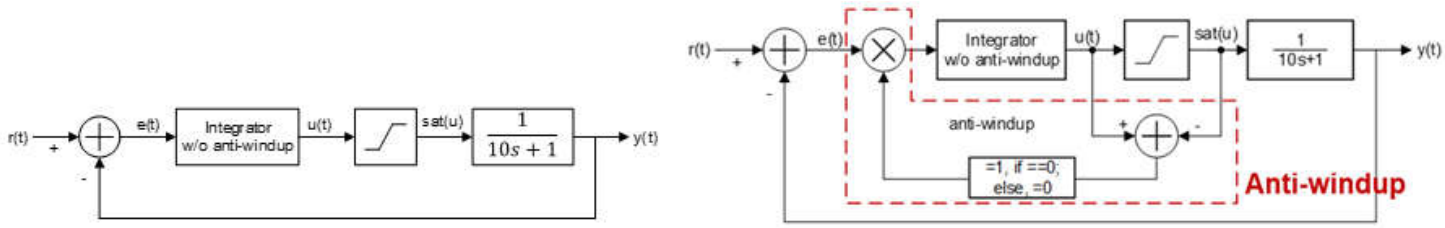


Figure 11. *Control diagram with and without anti-windup logic implementation.*

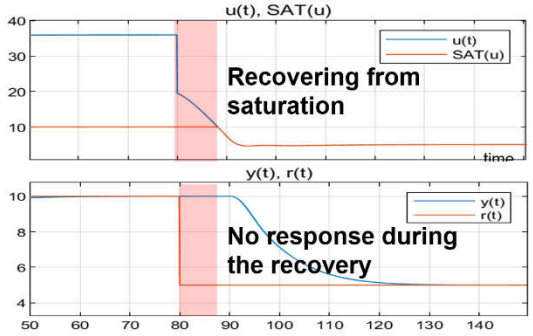


Figure 12. *Response with and without anti-windup logic.*

The following code snippets illustrate a 2P2Z controller with anti-windup implemented in C using the C2000 Digital Control Library (DCL). The original 2P2Z structure from ***Implementation*** can be split into two parts: an immediate term and a partial (pre-computed) integrator term. This separation makes it straightforward to apply integrator clamping and implement anti-windup. During the k^{th} sampling interval, the controller first computes the immediate term. The partial result is then pre-computed for use in the $(k + 1)^{th}$ interval. Because the control effort can be clamped right after the immediate term is calculated, updating the pre-computed integrator branch can be made conditional on the clamp result. If u(k) reaches the actuator limits, there is no need to update the partial integrator state for the next interval, and that computation can be skipped. This architecture also reduces control-loop latency. By partially pre-computing the control law, the "sample-to-output" delay (the time from sampling e(k) to producing u(k)) can be reduced to roughly one multiplication and one addition. The pre-computed structure allows u(k) to be applied to the actuator as soon as it is available. The remaining terms of the third-order control law do not depend on the newest input e(k), so they can be

calculated afterward without affecting u(k). As a result, input-output latency decreases and the controller responds more quickly.

```
// Executes an immediate 2nd order Direct Form 2 controller
static
            inline float32_t DCL_runDF22_C5(DCL_DF22 *p, float32_t ek)
{
    return((ek * p->b0) + p->x1);
}
// Executes a partial pre-computed 2nd order Direct Form 2 controller
static
            inline
            void
            DCL_runDF22_C6(DCL_DF22 *p, float32_t ek, float32_t uk)
{
    p->x1 = (ek * p->b1) + p->x2 - (uk * p->a1);
    p->x2 = (ek * p->b2) - (uk * p->a2);
}
// Saturates a control variable and returns 1 if either limit is exceeded
static
            inline int16_t DCL_runClamp_C2(float32_t *data, float32_t Umax, float32_t Umin)
{
    float32_t iv = *(data);
    *(data) = (*(data) > Umax) ? Umax : *(data);
    *(data) = (*(data) < Umin) ? Umin : *(data);
    return(((iv < Umax) && (iv > Umin)) ? 0 : 1);
}
// Pseudo code for implementing anti-windup which helps reducing control latency as well
// Runs a partial or integrator part only when control effort not saturated
uk = DCL_runDF22_C5(&df22, ek);               // Calculates immediate part of compensator
updateDuty(uk);                          // Update the actuator duty
s = DCL_runClamp_C2(&uk, upperLim, lowerLim); // Check if control effort is saturated
if (0U == s)
{
DCL_runDF22_C6(&df22, ek, uk);             // Do not run integrator if control saturated
}
```

Design Example: Voltage Mode Controlled Synchronous Buck Converter

This design example focuses on the control loop of a buck converter. The converter is assumed to operate in continuous conduction mode (CCM) and to be regulated using voltage-mode control (VMC). The design of the power stage is outside the scope of this discussion and is considered to be already complete, as shown in **Figure 13**. In this example, the input voltage is 48V, the output voltage is 12V with a maximum load current of 10A, the inductor is 33µH, and the output capacitor is 1060µF with an effective series resistance (ESR) of 10mΩ. With the power stage defined, the focus shifts to design the digital control loop, including the feedback circuitry, the ADC, the digital compensator and the DPWM actuator.

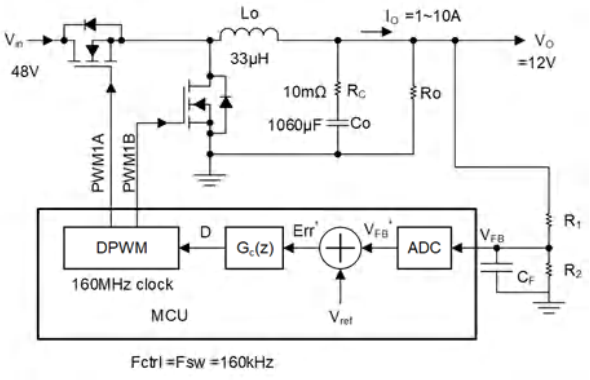


Figure 13. *Voltage mode control of buck converter.*

Feedback Design

First, let's design the feedback circuit for digital control as shown in **Figure 14**. The feedback scale is mainly decided by the full-scale-range of the ADC. For C2000 or a similar MCU, it is generally 3.3V. Considering the overvoltage protection (OVP) point at V_{OVP} = 18V, the maximum value of the feedback signal is given by **Equation 4**.

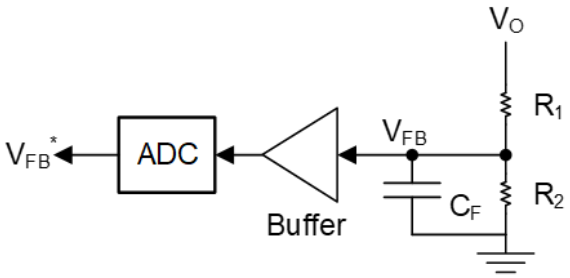


Figure 14. *Voltage mode control of buck converter.*

$$V_{FB} = V_O \times \frac{R_2}{R_1 + R_2} \tag{4}$$

If R_2 = 1kΩ then R_1 can be obtained from **Equation 5**.

$$R_1 = 1k\Omega \times \frac{18V - 3.3V}{3.3V} = 4.5k\Omega \tag{5}$$

Now, consider using an internal OPA as buffer in front of the ADC. In that case, the capacitor C_F before the ADC combine with R1 and R2 forms the anti-alias filter. The cut-off frequency can be calculated using **Equation 6**.

$$f_{AA} = \frac{1}{2\pi \times C_F \times (R1 \parallel R2)} = 40kHz \tag{6}$$

We set the anti-alias frequency at half of the Nyquist frequency, then C_F can be calculated using **Equation 7**.

$$C_F = \frac{1}{2\pi \times f_{AA} \times (R1 \parallel R2)} \cong 4.7nF \tag{7}$$

Now describe the feedback loop with the transfer function. Here we use unity '1' to present the full range of the ADC, so the feedback ratio K_V is obtained using **Equation 8** with unit '1' dividing the full range. It has the unit V^{-1}.

$$K_V = \frac{1.0(p.u.)}{18V} = 0.056V^{-1} \tag{8}$$

There is only one pole coming from the anti-alias filter, then we can write down the transfer function as shown in **Equation 9** from which will be used for obtaining the bode plot of feedback function H(s) in next sections.

$$H(s) = 0.056V^{-1} \times \frac{1}{1 + \frac{s}{2\pi \times 40kHz}} \tag{9}$$

Actuator Design

The next step in the design is to configure the actuator for executing the control effort. In this case, trailing-edge modulation is selected, with the control value reloaded at the start of each switching cycle. This sequence, illustrated in

Figure 15, demonstrates the order of events within the control loop. Under these conditions, the DPWM execution delay equals one duty cycle period, which is calculated as 1.563µs (see **Equation 10**). This delay introduces additional control latency and reduces the available phase margin, making it an important factor to consider in the design.

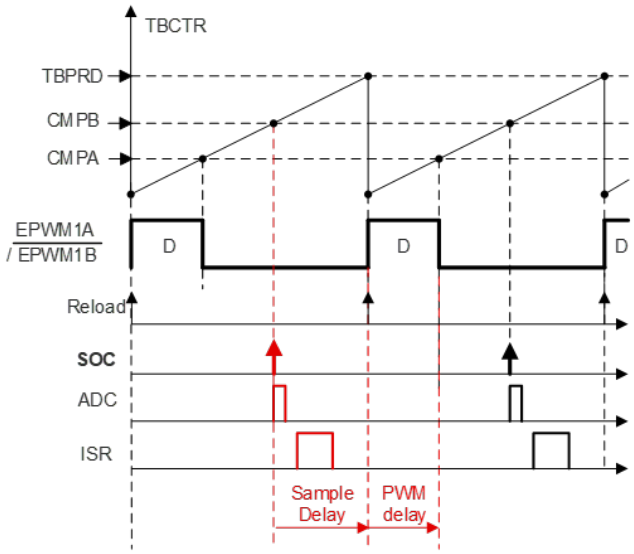


Figure 15. *Sequence of events from sampling instance to duty update.*

So, the PWM execution delay equals to the duty cycle period. Here, it is calculated as 1.563µs as shown in **Equation 10**.

$$\mathrm{Td_{PWM}} = \mathrm{Duty} \times \mathrm{T_s} = \frac{12\mathrm{V}}{48\mathrm{V}} \times \frac{1}{160\mathrm{kHz}} = 1.563\mu\mathrm{s} \tag{10}$$

Unlike analog PWM, digital PWM (DPWM) allows the duty cycle to be used directly as the actuator input, with the compare value computed during PWM execution. This approach removes the dependence of the control loop on the period coefficient. In addition to PWM delay, ADC conversion also contributes to control latency. The ADC delay spans from the start-of-conversion (SOC) trigger to the point when the conversion result is reloaded. To minimize this delay, both the ADC conversion time and the interrupt service routine (ISR) execution time must be carefully estimated. The ADC conversion time consists of the sample-and-hold period and the actual bit conversion time, both of which are configurable and can be obtained from the MCU technical reference manual. Estimating ISR execution time is more difficult because it includes compensator calculations as well as the overhead of entering and exiting the ISR. For robust design, the longest possible ISR time should be assumed, which can be determined through ISR profiling. In this example, the worst-case delay is approximated as 0.65µs. Using this estimate, the CMPB is configured to trigger the SOC event 0.7µs before the PWM reload, as represented in **Equation 11**.

$$\mathrm{Td_{ISR}} = 0.7\mu s \tag{11}$$

Now, we can get the total control delay by summing the PWM delay and ADC delay as shown in **Equation 12**.

$$\mathrm{T_d} = \mathrm{Td_{PWM}} + \mathrm{Td_{ISR}} = 2.263\mu\mathrm{s} \tag{12}$$

The transfer function of this delay can be expressed using **Equation 13**.

$$\mathrm{G_d}(s) = \mathrm{e}^{-s \times \mathrm{T_d}} = \mathrm{e}^{-s \times 2.263\mu\mathrm{s}} \tag{13}$$

Plant Transfer Function

Now let us include the power stage in the control loop analysis. For this purpose, we focus only on the transfer function from duty cycle to output voltage, as this relationship is essential for calculating the loop gain and designing the compensator. In a buck converter operating under VMC in continuous conduction mode (CCM), two important corner frequencies are present. The first is a complex double pole, which arises from the interaction between the output inductor and capacitor. The second is an ESR zero, introduced by the equivalent series resistance (ESR) of the output capacitor. The resulting transfer function is expressed as shown in **Equation 14**.

$$G_{vd}(s) = \frac{\hat{v}_o}{\hat{d}} = V_{in} \times \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{Q \times \omega_0} + \frac{s^2}{{\omega_0}^2}} \tag{14}$$

where

- $G_{vd}(s)$ is the gain of the transfer function from D to V_O
- V_{in} is the input voltage
- ω_O is a complex double pole that comes from the inductor (L_O) and output capacitor (C_O), and can be calculated using **Equation 15**.

$$\omega_0 = \frac{1}{\sqrt{LoCo}} = 5.347\text{kHz} \tag{15}$$

- ω_Z is ESR zero comes from ESR (Rc) of the output capacitor, the capacitance of the clamping capacitor, and can be calculated from **Equation 16**.

$$\omega_z = \frac{1}{R_C Co} = 94.34\text{kHz} \tag{16}$$

From **Equation 14**, the corresponding bode plot can be obtained, as shown in **Figure 16**. Both the plot and the equation reveal that the load condition directly influences the quality factor (Q). A higher Q value introduces phase distortion around the double pole frequency. Therefore, it is important to evaluate the phase margin under the lightest load condition, where the Q value is typically highest. If the Q is excessively large, the phase margin may fall well below 45 degrees near the double pole, making it very difficult to achieve stable compensation at loop bandwidths above this frequency.

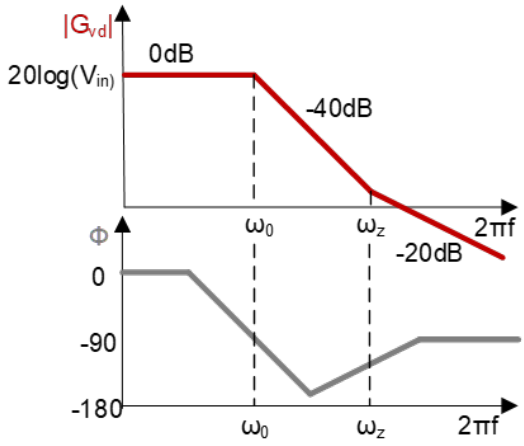


Figure 16. *Bode plot of the transfer function from D to Vo.*

Now, everything is in the control loop. The plant transfer function before compensation is the multiplication of the individual transfer function of the power stage $G_{vd}(s)$, feedback loop H(s), and control delay $G_d(s)$ and can be calculated using **Equation 17**.

$$G_{plant}(s) = G_{vd}(s) \times H(s) \times G_d(s) \tag{17}$$

By putting the values of **Equation 9**, **Equation 13**, and **Equation 14** into **Equation 17**, the plant transfer function without compensation can be represented as **Equation 18**.

$$G_{plant}(s) = V_{in} \times \frac{1 + \frac{s}{\omega_z}}{1 + \frac{s}{Q \times \omega_0} + \frac{s^2}{{\omega_0}^2}} \times \frac{K_V}{1 + \frac{s}{\omega_{AA}}} \times e^{-s \times Td} \tag{18}$$

From **Equation 18**, a complete plant bode plot can be obtained as shown in **Figure 17**.

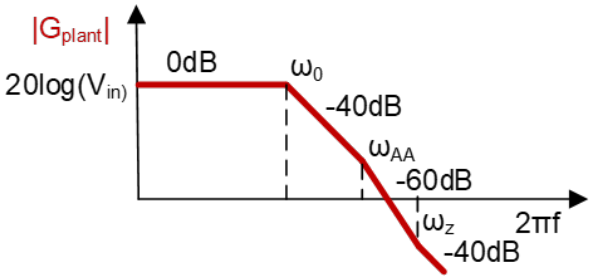


Figure 17. *Bode plot of plant transfer function.*

Compensator Design

The next step is to design the compensator by adding poles and zeros to shape the loop response of the plant. In this example, a 2P2Z compensator is selected, although alternatives such as PI or PID can also be used. The 2P2Z structure is chosen because it allows one zero-pole pair to provide infinite DC gain, ensuring zero steady-state error, while two additional zeros can be positioned to cancel the effect of the double pole, as expressed in **Equation 19**.

$$\omega_{z1} = \omega_{z2} = \omega_0 = 5.347\text{kHz} \tag{19}$$

To address the ESR zero, one additional pole is introduced, as shown in **Equation 20**. In practice, however, the double pole cannot be completely canceled in this way, as the effectiveness of compensation depends heavily on the Q factor of the power stage.

$$\omega_{p1} = \omega_z = 94.34\text{kHz} \tag{20}$$

Next, the crossover frequency must be selected. As a general guideline, it is set at approximately one-tenth of the switching frequency, as expressed in **Equation 21**. In digital implementations, the crossover frequency can be set lower, often down to one-twentieth of the switching frequency.

$$\omega_c = 2\pi f_c = 2\pi \times 16\text{kHz} \tag{21}$$

At the chosen crossover frequency, the plant gain can be obtained from the transfer function. Multiplying this gain with the compensator gain should equal unity, which allows the K_{DC} coefficient to be calculated using **Equation 22**.

$$K_{DC} = \frac{\omega_c\left(1+\frac{\omega_c}{\omega_{p1}}\right)}{\left(1+\frac{\omega_c}{\omega_{z1}}\right)\left(1+\frac{\omega_c}{\omega_{z2}}\right)} \times \frac{1}{G_{plant}(\omega_c)} = 40.374\text{kHz} \tag{22}$$

In practice, these design calculations are typically performed with software tools such as Mathcad or MATLAB. **Figure 18** illustrates the Bode plot of the uncompensated plant, the 2P2Z compensator, and the resulting closed-loop response

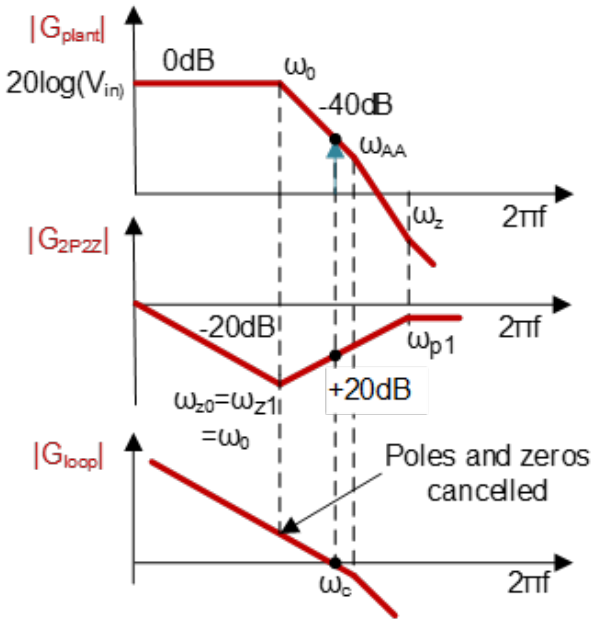


Figure 18. *Bode plot of plant, compensator, and loop transfer function.*

Finally, before closing the control loop design, stability must be verified by calculating the phase margin. As a general rule, a phase margin greater than 45 degrees is required for stable operation. If the margin is insufficient, stability can be improved by slightly reducing the compensator zero locations, increasing the compensator pole frequency, or lowering the overall loop bandwidth. In this example, the calculated phase margin is 49 degrees, as shown in **Equation 23**, which is adequate for stable operation.

$$\mathrm{PM}\left(\omega_c\right) = 180° + \mathrm{atan}\left(\frac{\mathrm{Im}G_{loop}(\omega_c)}{\mathrm{Re}G_{loop}(\omega_c)}\right) = 49.3° \tag{23}$$

Up to this point, the analysis has been carried out in the s-domain, corresponding to analog representations. However, the actual compensator must be implemented in digital form. Therefore, the poles and zeros determined in the previous steps must be converted into z-domain coefficients using the trapezoidal (Tustin) approximation. It should be noted that this conversion is not exact—accuracy decreases as the operating frequency approaches half of the control frequency (the Nyquist frequency). **Table 4** summarizes the conversion process, providing the equations and the resulting digital coefficients for the 2P2Z compensator. These coefficients are then implemented in the control code, as described in ***Anti-windup for Integrator***, which also includes the anti-windup mechanism

Domain	Analog	Digital
Transfer Function	See **Equation 24**	See **Equation 25**
Coefficients	K_{DC} = 40.374kHz $\omega_{p1} = \omega_z$ = 94.34kHz $\omega_{z1} = \omega_{z2} = \omega_0$ = 5.347kHz	b_0 = 106.367 b_1 = –205.742 b_2 = 99.49 a_1 = –1.545 a_2 = –0.545

Table 4. *Analog 2P2Z compensator coefficients and the digital approximation.*

$$G_{2p2z}(s) = \frac{K_{DC}}{s} \times \frac{\left(1 + \frac{s}{\omega z1}\right)\left(1 + \frac{s}{\omega_{z2}}\right)}{\left(1 + \frac{s}{\omega_{p1}}\right)} \tag{24}$$

$$\frac{u(k)}{e(k)} = \frac{b_0 + b_1 \times z^{-1} + b_2 \times z^{-2}}{1 + a_1 \times z^{-1} + a_2 \times z^{-2}} \tag{25}$$

Conclusion

Although the digital control loop design procedure presented in this paper is demonstrated using a buck converter, the same approach can be applied to other duty-controlled topologies such as boost converters and four-switch buck-boost converters. It can also be extended to dual-loop implementations, such as average current mode control used in power factor correction for AC-DC conversion. In such cases, the innermost current loop is designed first, and the same methodology is then applied to the outer voltage loop, treating the inner loop as the plant. These techniques can further be adapted for variable frequency-controlled topologies, such as resonant converters like the LLC, as well as for variable phase-controlled topologies, including the phase-shifted full bridge. More broadly, MCU-based digital power control provides the flexibility to handle advanced power conversion systems. It enables the implementation of multi-mode, multi-phase, multi-level, and interleaved topologies, along with the associated complex control algorithms, thereby offering superior adaptability and performance across a wide range of applications.

References

1. Texas Instruments, **Control Theory Seminar, Training video**
2. Texas Instruments, ***Control Theory Workshop, Student Manual***
3. Texas Instruments, ***Real-Time Control Reference Guide***
4. Texas Instruments, ***Precision labs series: Analog-to-digital converters (ADCs), Training video***
5. Texas Instruments, ***C2000 Academy***
6. Texas Instruments, ***C2000 Digital Control Library (DCL) User's Guide***
7. Texas Instruments, ***Understanding And Applying Current-mode Control Theory Application Report***
8. Texas Instruments, **TI Digital Power Solutions**
9. Oppenheim, A. V., Willsky, A. S., and Nawab, S. H. (1997). Signals & Systems (2nd ed.). Prentice Hall

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale, TI’s General Quality Guidelines, or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Last updated 10/2025