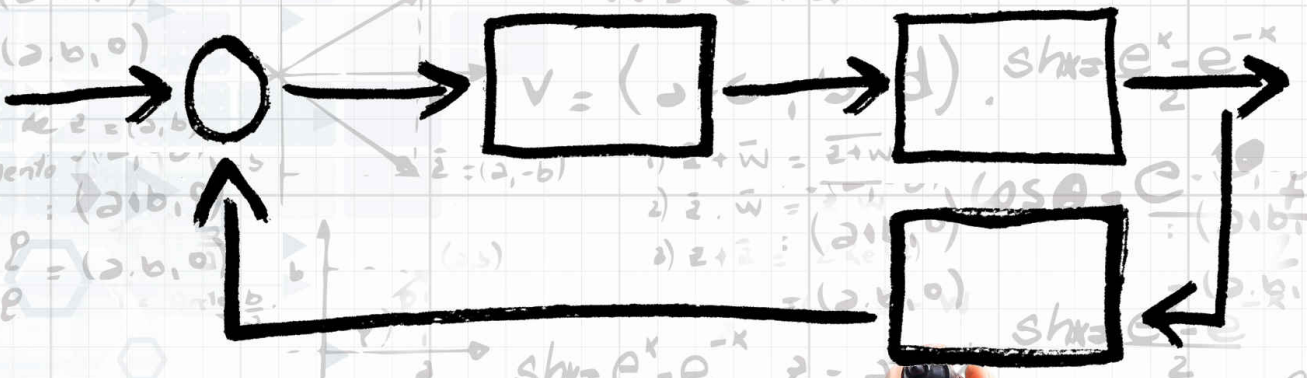
Real-Time Control Reference Guide

Texas Instruments



Message from the editors

This document is intended as a valuable quick guide for often used system-level design formulae and real-time control concepts in order to help facilitate real-time control application design. We hope you find this collection of material useful.

Here is a brief overview of the key areas included:

- Mathematical Models
- First and Second Order Systems
- Filters
- Controller Types
- A/D Conversions
- Comparator Basics
- Characteristics of a Real-Time Processor
- Encoder Basics
- Pulse Width Modulation Basics
- D/A Conversions

Additional resources to explore:

Control Theory Seminar

- A four-part technical seminar that offers an introduction to control theory covering fundamental concepts, feedback systems, transient response, and discrete-time systems.

State Space Control Seminar

- A four-part course in control theory based on the state space modeling paradigm covering state space models, properties of linear systems, state feedback control, and linear state estimators

Introduction to Microcontroller Programming for Power Electronics Control Applications

- This book contains the fundamental subjects of the interdisciplinary field of power electronic based systems, which draws knowledge from circuit and control theory, (digital) signal processing for embedded implementation electrical machines/drives and power semiconductor devices. This book also presents state-of-the-art techniques to implement modulation schemes and control algorithms in a commercial microcontroller (MCU) suitable for rapid prototyping approach, and hint for designing analog circuits, such as low voltage converter, output filters/load.

Control Theory Fundamentals

- This book provides a readable introduction to control of both continuous time and discrete time systems. The first four chapters of the book cover classical methods using transfer functions. The remaining chapters cover analysis and design using state space methods. Worked examples are included to illustrate key topics in each section. The book contains five appendices; a review of matrix algebra, reference tables of Laplace and z transforms, supporting Matlab scripts, and a case study in controller design using state space methods.

Digital Control of Dynamic Systems

- This book's emphasis is on designing digital controls to achieve good dynamic response and small errors while using signals that are sampled in time and quantized in amplitude. Both transform (classical control) and state-space (modern control) methods are described and applied to illustrative examples.

Digitally Controlled High Efficiency and High Power Density PFC Cicuits - 3 Part Series

- These series of presentations introduce two bridgeless PFC designs using C2000™ MCU. TI high voltage GaN is used to implement a 3.3kW interleaved CCM totem-pole PFC and a 1.6kW interleaved TRM totem-pole PFC designs. Detailed design considerations are provided to minimize switching loss, current crossover distortion, input current THD and improve efficiency and PF.

Digital Power SDK

- C2000's Digital Power SDK is a cohesive set of software infrastructure, tools, and documentation designed to minimize C2000™ MCU-based digital power system development time targeted for various AC-DC, DC-DC and DC-AC power supply applications

TIDM-DC-DC-BUCK

- The BOOSTXL-BUCKCONV reference design provides a quick and easy way to learn about digital power supply control and design using C2000 devices.

TI Reference Designs

- Ready-to-use reference designs with theory, calculations, simulations, schematics, PCB files and bench test results.

TI Precision Labs

- On-demand courses and tutorials ranging from introductory to advanced concepts that focus on application-specific problem solving

TI E2E Community

- Support forums for all TI products

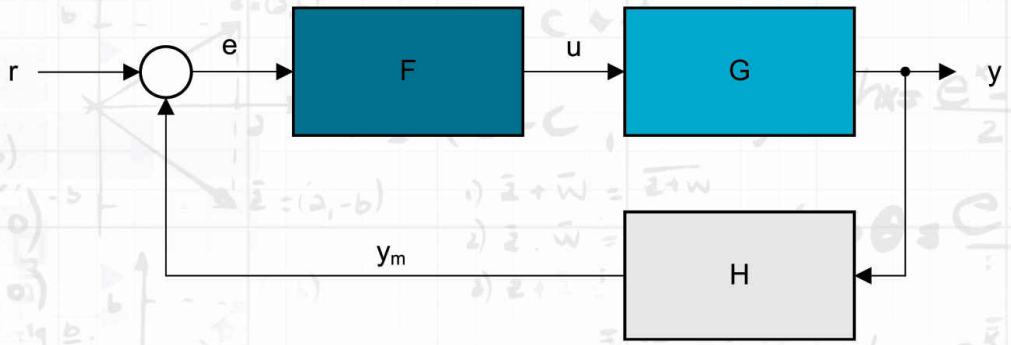
Table of Contents

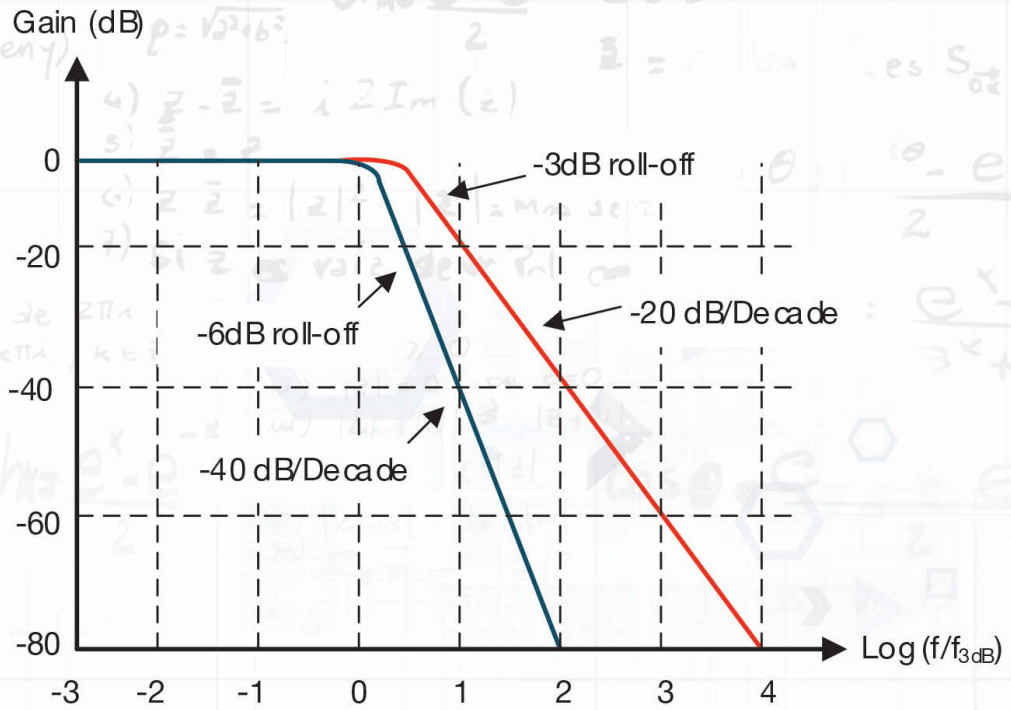
List of Figures

List of Tables

System Design

- Control
- Feedback control
- Dynamic systems
- System stability
- Timing requirements
- Discrete Time Domain
- Filters
- Notes





Control

"A control system is considered to be any system which exists for the purpose of regulating or controlling the flow of energy, information, money, or other quantities in some desired fashion" - *W.L.Brogan, Modern Control Theory, 1991*.

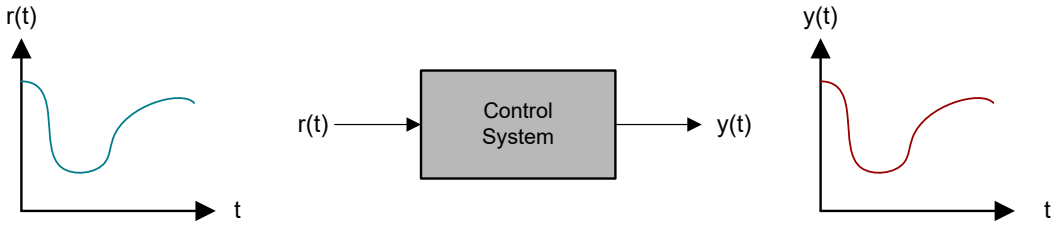


Figure 1. *Control system.*

The principle objective of control is to force the system output to accurately follow an input.

Table 1. *Characteristics of a good control system.*

Term	Definition/Explanation
Stability	**Stability** is the quality, state, or degree of being stable. In order for a system to be stable, the gain and phase margins need to be positive.
Steady State Accuracy	How close the output is to the desired input. The closer the output is to the input the better the **steady state accuracy** is.
Satisfactory Transient Response	A **transient response** shows how the system behaves when it is taken out of equilibirum. Ideally, the system should not deviate from the steady state.
Satisfactory Frequency Response	A **frequency response** displays the output of a system in terms of magnitude and phase as a function of frequency.
Reduced Sensitivity to Disturbances	Disturbance in control systems signify any unwanted or unexpected inputs that alter the system's output which cause an increase in the **system error**.

Open loop versus closed loop

There are two types of control systems, open and closed. Within an **open control system**, the control action is not dependent on external influences nor does it incorporate feedback to alter the output response. In contrast, within a **closed loop system** the control action is indeed dependent on external influences and utilizes feedback to adapt and achieve the desired results.

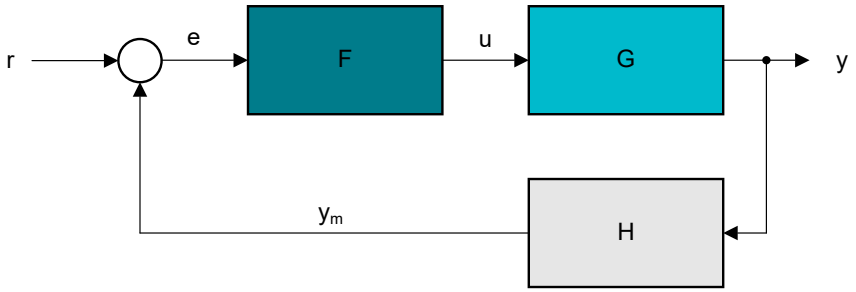


Figure 2. *Closed loop system.*

Where

F = controller transfer function

G = plant transfer function

H = sensor transfer function

r = reference input

e = error signal

u = control effort

y = output

y_m = feedback

Equations:

Error equation

$$e = r - Hy \tag{1}$$

Output Equation

$$y = FGe \tag{2}$$

Error and output equations combined

$$y\,(1 + FGH) = FGr \tag{3}$$

Open loop **transfer function**

$$L = FGH \tag{4}$$

Closed loop **transfer function**

$$\frac{y}{r} = \frac{FG}{1 + FGH} \tag{5}$$

Feedback control

Feedback is when any part of the system's output is brought back to the input and utilized as part of the input to the system. It is sometimes referred to as "closed loop control". Feedback control can be categorized as one of the most important segments within any closed loop system since it improves the performance of the system.

When properly applied feedback can:

- Reduce or eliminate steady state error characteristics.
- Reduce the sensitivity of the system to parameter variations.
- Change the gain or phase of the system over some desired frequency range.
- Reduce the effects of load disturbance and noise on system performance.
- Cause an unstable system to become stable.

- Linearize a non-linear component.

Error ratio

The error ratio (sensitivity function) determines the loop sensitivity to disturbance.

$$\frac{e}{r} = \frac{1}{1 + FGH} = \frac{1}{1 + L} = S \tag{6}$$

Dynamic systems

First order system

Equations

1st order differential equation

$$\tau y'(t) + y(t) = u(t) \tag{7}$$

Transfer function of 1st order system

$$\frac{y(s)}{u(s)} = \frac{1}{s\tau + 1} \tag{8}$$

Response of system

$$y(t) = L^{-1}\left\{\frac{1}{s\tau + 1}u(s)\right\} \tag{9}$$

Response of system following a unit step response

$$y(t) = 1 - e^{-\frac{t}{\tau}} \tag{10}$$

Where

τ = time constant of the system

$y(t)$ = output function

$u(t)$ = input function

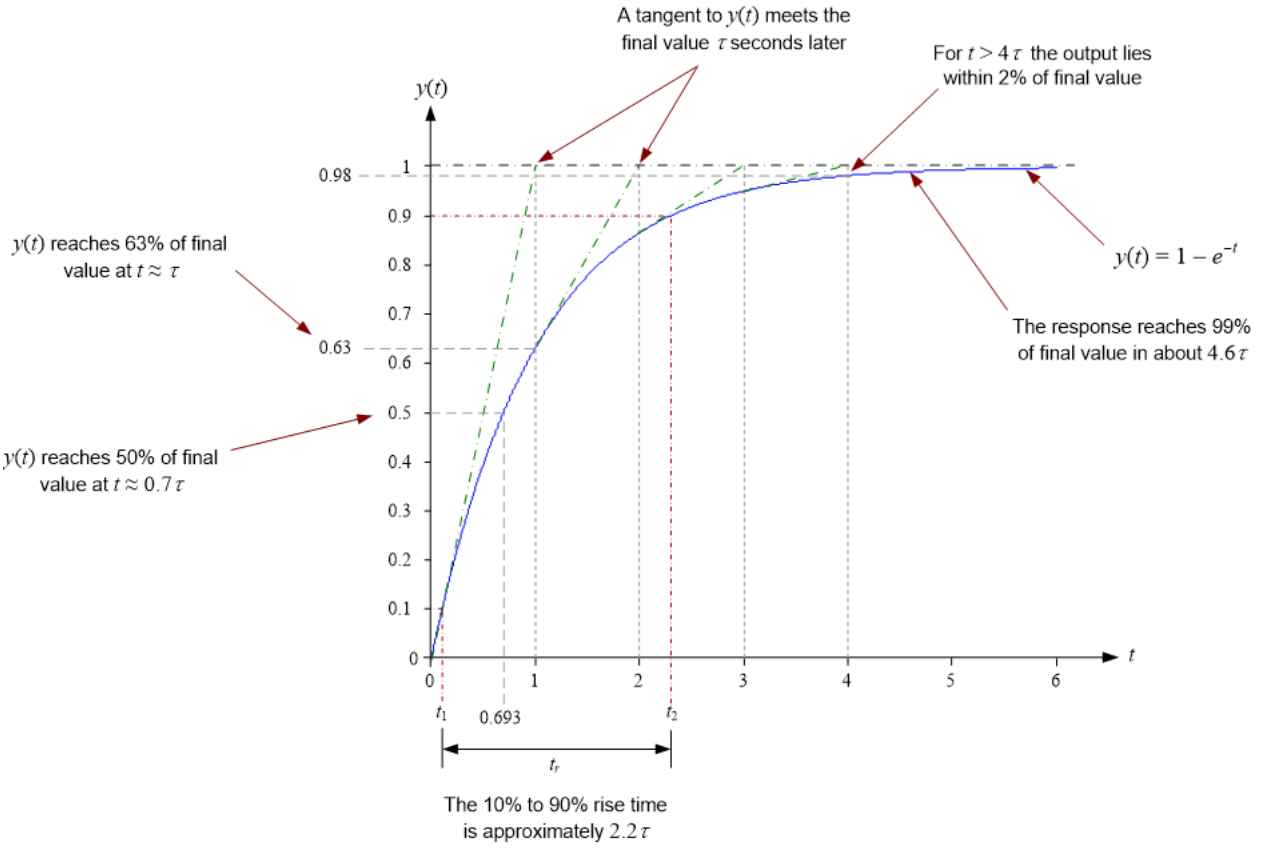


Figure 3. *Unit step response of a 1st order system (τ=1).*

Second order system

Equations

2nd order differential equation

$$y''(t) + 2\zeta\omega_n y'(t) + \omega_n^2 y(t) = \omega_n^2 u(t) \tag{11}$$

Transfer function of 2nd order system

$$\frac{y(s)}{u(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{12}$$

Characteristic equation

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \tag{13}$$

Poles of a linear 2nd order system

$$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} \tag{14}$$

Unit step response of an under-damped 2nd order system

$$y(t) = 1 - \frac{\omega_n}{\omega_d} e^{-\sigma t} \sin(\omega_d t + \varphi) \quad (15)$$

$$\varphi = \cos^{-1} \zeta \quad (16)$$

Where

$y(t)$ = output function

$u(t)$ = input function

ω_n = un-damped natural frequency

ω_d = damped natural frequency

ζ = damping ratio

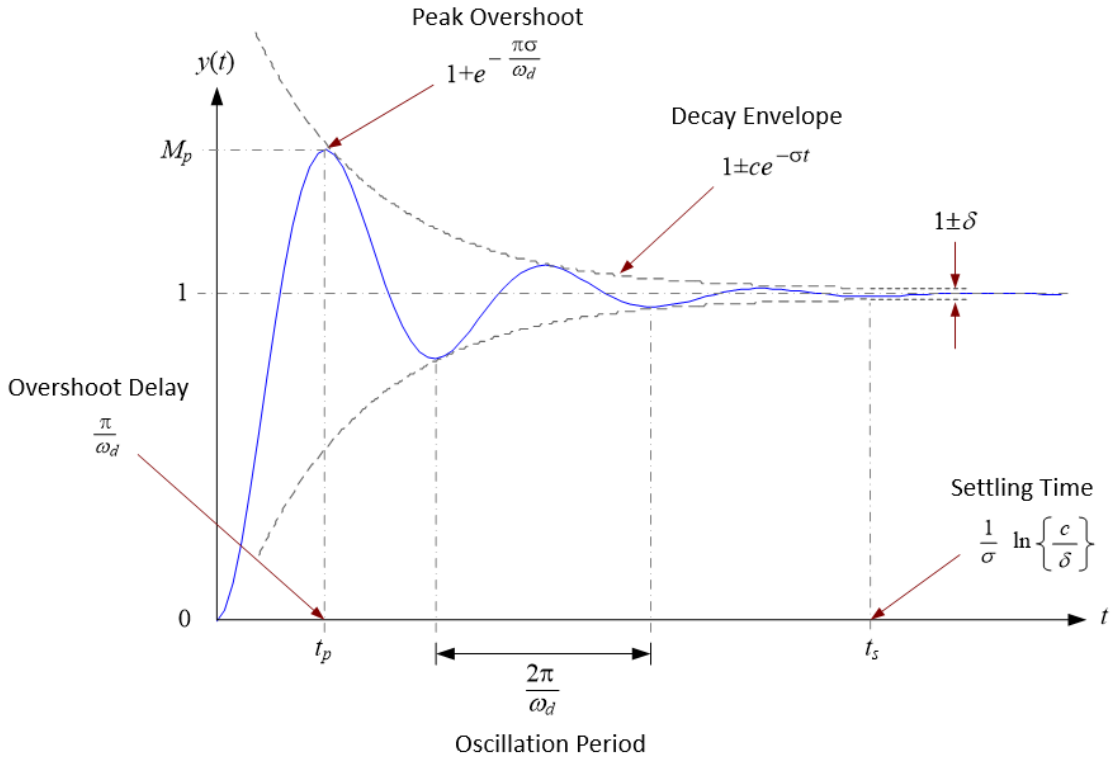


Figure 4. *Unit step response of an under-damped 2nd order system.*

System stability

Stability is one of the most important requirements when designing a control system. If the **closed loop transfer function** of a system is known then the stability of the system can be observed through the denominator of the transfer function in factored form (**characteristic equation**) by identifying if the real parts are positive or negative. However, in order to accurately measure how stable a system is, the gain and phase margins are used. Both of these margins are obtained through the **Bode plot** of the frequency response. Gain and Phase margins define the amount of change in open-loop gain and phase that is required to make a closed-loop system stable.

For a system to be stable, both the gain and phase margins need to be positive.

Gain margin

The gain margin is the difference between 0 dB and the gain at the phase cross-over frequency that gives a phase of –180°. If the gain margin $|GH(j\omega)|$ at the frequency of $\angle GH(j\omega) = -180°$ is greater than 0 dB (positive gain margin), then the closed-loop system is stable.

Phase margin

The phase margin is the difference in phase between –180° and the phase at the gain cross-over frequency that gives a gain of 0 dB. If the phase $\angle GH(j\omega)$ at the frequency of $|GH(j\omega)| = 1$ is greater than –180° (positive phase margin), the closed-loop system is stable.

Phase margin is related to the **damping** of a second-order system through the following equation:

$$\zeta = \frac{PM}{100} \tag{17}$$

Where

ζ = damping ratio

PM = phase margin

Timing requirements

Peak/rise time

Table 2. *Rise and peak time for systems.*

System Order	System Classification	Rise Time %	Peak Time
1st Order	-	10 - 90	Usually categorized as 2.2 τ, where τ is the time constant
2nd Order	Under-damped	0 - 100	Time at which the step response reaches its maximum
2nd Order	Over-damped	10 - 90	Peak time is not defined

Equation

Peak Time

$$t_p = \frac{\pi}{\omega_n \beta} = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}} \tag{18}$$

Where

ζ = damping ratio

ω_n = undamped natural frequency

Settling time

The settling time is the time required for a system to settle within a certain percentage of the input amplitude. For a second order system, a settling time is desired in which the response remains within 2% of its desired value.

Definitions

Settling time of **1st order system**

$$4\,\tau \tag{19}$$

Settiling time of **2nd order system**

$$t_s = \frac{1}{\sigma}\ln\left(\frac{c}{\delta}\right) \tag{20}$$

$$c = \frac{\omega_n}{\omega_d} \tag{21}$$

Damping coefficient

$$\sigma = \zeta\omega_n \tag{22}$$

Lograrithmic decrement

$$\delta = \ln\left(\frac{x_0}{x_1}\right) \tag{23}$$

Tolerance fraction for underdamped systems (in most cases this is 0.02)

$$\frac{c}{\delta} = \sqrt{1-\zeta^2} \tag{24}$$

Where

τ = time constant (time it takes for the step response to reach 63% of its final value)

x_0/x_1 = the amplitudes of two successive peaks within a step response

ω_n = un-damped natural frequency

ω_d = damped natural frequency

ζ = **damping** ratio

Overshoot

Overshoot quantifies the amount the step response deviates from the ideal settling amplitude at the peak time.

Definitions

Peak response

$$M_p = 1 + \exp\left(\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}\right) \tag{25}$$

Percentage of overshoot

$$P.O. = 100e^{-\pi\zeta/\sqrt{1-\zeta^2}} \tag{26}$$

Damping ratio

$$\zeta = \frac{-\ln\left(\frac{PO}{100}\right)}{\sqrt{\pi^2 + \ln^2\left(\frac{PO}{100}\right)}} \tag{27}$$

Damping

The dynamic behavior of a **second order system** is defined by the damping ratio and un-damped natural frequency.

Table 3. *Damping ratios based on classification.*

Damping Ratio	Roots	Classification
$\zeta > 1$	$s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$	Over-damped
$\zeta = 1$	$s = -\omega_n$	Critically damped
$0 < \zeta < 1$	$s = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$	Under-damped
$\zeta = 0$	$s = \pm j\omega_n$	Un-damped

Where

ζ = damping ratio

ω_n = un-damped natural frequency

Delay

There are many types of delays that can be introduced within a control system, some are inherent to the system and some are external.

Definitions

Delay in the time domain

$$\mathcal{L}\{f(t - T)\} \tag{28}$$

Delay in **laplace domain**

$$e^{-sT}f(s) \tag{29}$$

Magnitude of delay

$$\left| e^{-i\omega T} \right| = \sqrt{(\cos \omega T)^2 + (-\sin \omega T)^2} = 1 \tag{30}$$

Phase of delay

$$< e^{-i\omega T} = \tan^{-1}\left(\frac{-\sin\omega T}{\cos\omega T}\right) = -\omega T \tag{31}$$

The time delay will not affect the magnitude since the magnitude of a pure delay is always equal to one, no matter the frequency of the input. However, the phase will get more and more negative as the frequency increases. If the phase becomes negative then the system could become unstable.

One way to compensate for a negative phase margin is to decrease the bandwidth but this will lead to slower performance. Which is why time delay in a control loop is detrimental both to **performance and stability**.

Delay Type	Closed Loop Transfer Function with Delay Included	Description
Sensor	$\frac{y}{r} = \frac{FG}{1 + FGe^{-sT}}$	The sensor that measures the process output, y, delays passing on the measured value by T time units
Actuator	$\frac{y}{r} = e^{-sT}\frac{FG}{1 + FGe^{-sT}}$	The input can influence the plant without passing through the delay

Where

e^{-sT} = delay transfer function

Discrete Time Domain

Discrete time refers to variables as occurring at distinct, separate "points in time". A discrete-time signal is a time series that consists of a sequence of quantities and is obtained by **Figure 6** from a continuous-time signal at uniformly spaced times.

Pros of Digital Control

- High repeatability and reliability
- Easy to prototype and make changes

Cons of Digital Control

- Increased '**delay**' in the feedback loop
- Lower performance compared to analog control

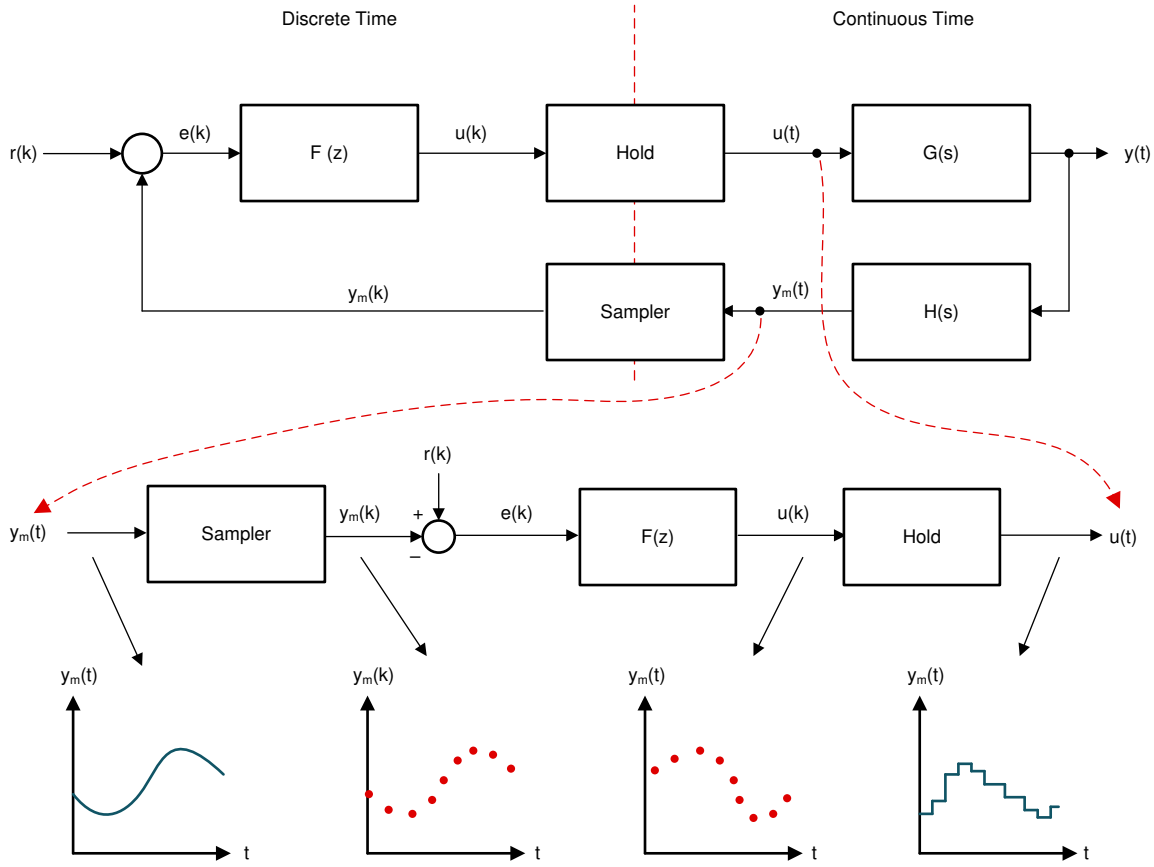


Figure 5. *Digital control system.*

Definitions:

- **Sample:** Signals represented as sequences of numbers.
- **Sampling Period (T):** The space between samples.
- **Sampling Frequency (F_T):** Inverse of the sampling period.

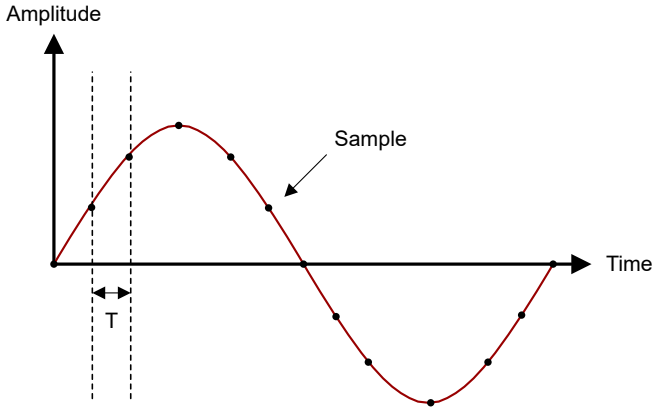


Figure 6. *Sampling.*

Note

In practice, continuous time signals are sampled using an **ADC**

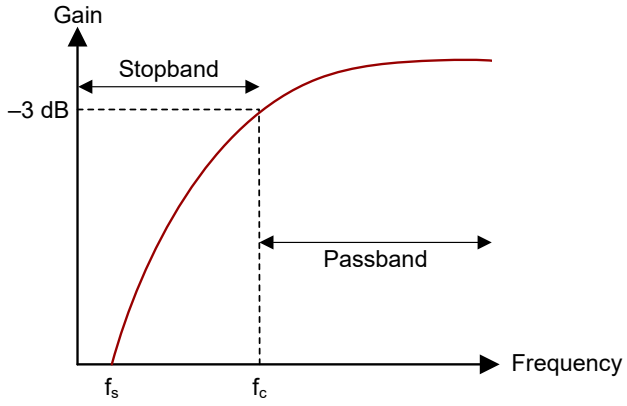
Filters

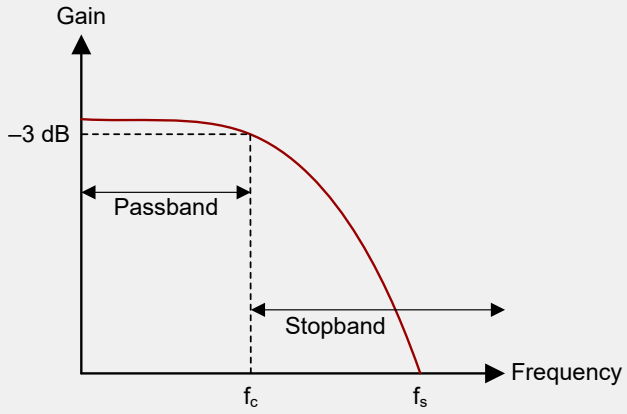
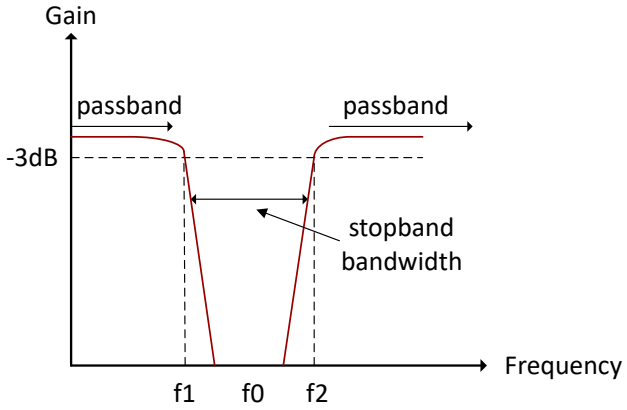
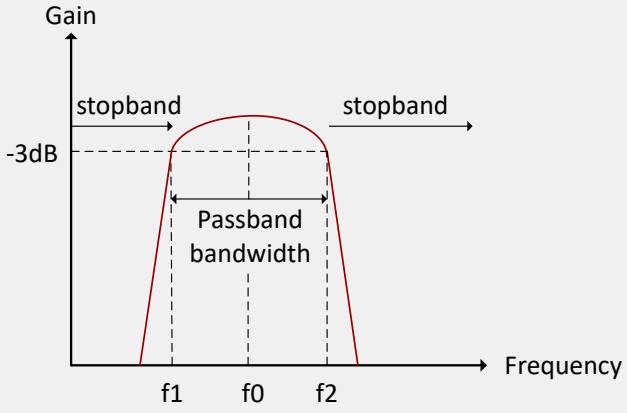
Definitions:

- **-3dB Frequency/Cutoff Frequency (f_{-3dB}/f_c):** The input frequency that causes the output signal to drop by -3dB relative to the input signal. For a high-pass and low-pass filter there is only one -3dB frequency, but for band-pass, and band-stop there are two -3db frequencies, referred to as f1 and f2 within **Table 4**.
- **Center Frequency (f_0):** This is a term that describes the central frequency that lies between the upper and lower cutoff frequencies of band-pass and band-stop filters.
- **Stopband Frequency (f_S):** A particular frequency at which the attenuation reaches a specified value. For low-pass and high-pass filters the frequencies beyond the stopband frequency are reffered to as the stopband. However, for pass-band or stop-band, and notch filters two stopbands exist.
- **Bandwidth (β):** The bandwidth is the width of the passband, and the passband is the band of frequencies that do not experience significant attenuation when moving from the input of the filter to the output of the filter
- **Quality Factor (Q):** The quality factor of a filter conveys its damping characteristics. In the time domain, damping corresponds to the amount of oscillation in the system's step response. In the frequency domain, higher Q corresponds to more (positive or negative) peaking in the system's magnitude response. For a band-pass, band-stop, and notch filter, Q represents the ratio between the center frequency and the -3dB bandwidth, $Q = f_0/(f_2 - f_1)$

Filter Types

Table 4. *Filter types.*

Type	Definition	Response Curve
High-Pass (HPF)	Removes low frequencies and retains high frequencies	

Type	Definition	Response Curve
Low-Pass (LPF)	Removes high frequencies and retains low frequencies	
Band-Stop (BSF)	Removes an interval of frequencies with a band, retains others	
Band-Pass (BPF)	Retains an interval of frequencies with a band, removes others	

Filter Orders

Filters can be of first order, second order, third order, and so forth. This is defined by the number of poles found within the **Section 10.2**. This chapter will focus on 1st and 2nd order filters. **Figure 7** shows the difference on a magnitude plot between a 1st (red curve) and 2nd order filter (blue curve).

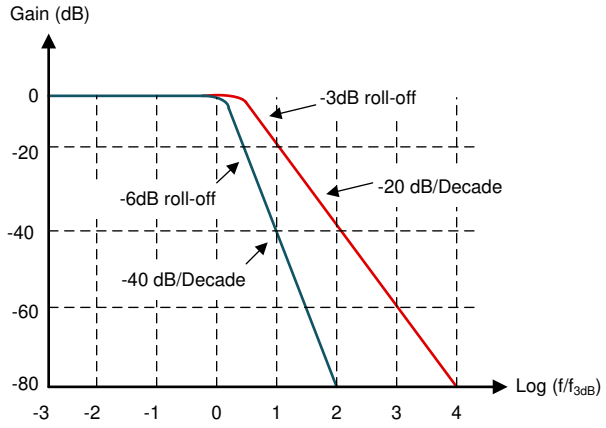
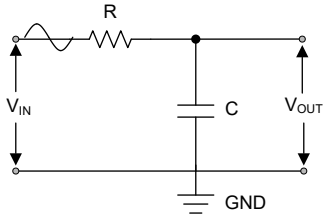
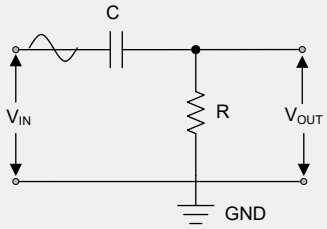
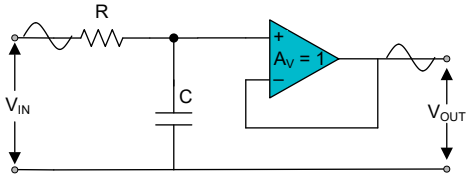


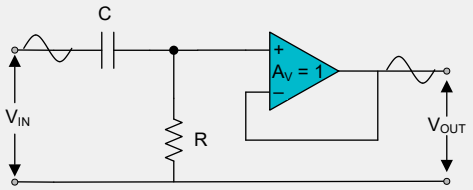
Figure 7. *Filter response curves.*

Definitions:

- **Passive:** Only includes passive components such as resistors, capacitors, and inductors.
- **Active:** Includes active components such as op-amps while still utilizing resistors and capacitors.

Table 5. *First order filters.*

Filter Type	Active/Passive	Circuit	Gain (Vout/Vin)
Low-Pass	Passive		$\frac{X_C}{\sqrt{R^2 + X_C{}^2}}$
High-Pass	Passive		$\frac{R}{\sqrt{R^2 + X_C{}^2}}$
Low-Pass	Active		$\frac{X_C}{\sqrt{R^2 + X_C{}^2}} = 1$

Filter Type	Active/Passive	Circuit	Gain (Vout/Vin)
High-Pass	Active		$\frac{R}{\sqrt{R^2 + X_C{}^2}} = 1$

Where

Transfer function of 1st order low pass filter

$$H(s) = \frac{\omega_C}{s + \omega_C} \quad (32)$$

Transfer function of 1st order high pass filter

$$H(s) = \frac{s}{s + \omega_C} \quad (33)$$

Cutoff frequency

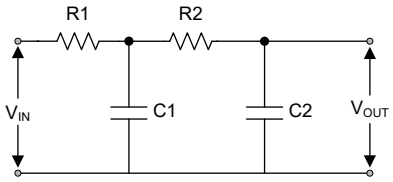
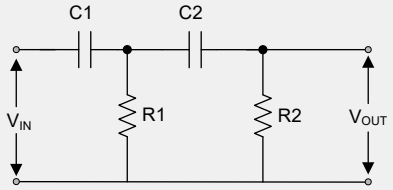
$$f_C = \frac{1}{2\pi RC} \quad (34)$$

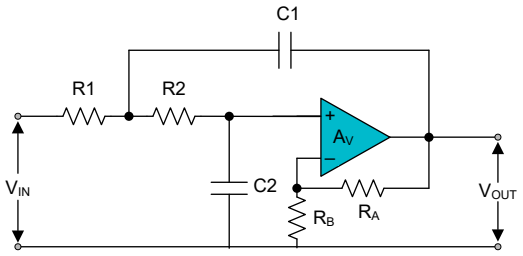
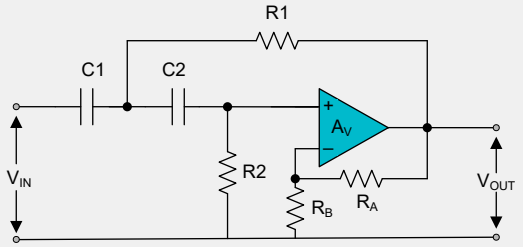
$$\omega_C = 2\pi f_C \quad (35)$$

Capacitive reactance

$$X_c = \frac{1}{2\pi fC} \quad (36)$$

Table 6. *Second order filters.*

Filter Type	Active/Passive	Circuit	Gain (Vout/Vin)
Low-Pass	Passive		$\left(\frac{1}{\sqrt{2}}\right)^n$ where n is the number of stages
High-Pass	Passive		

Filter Type	Active/Passive	Circuit	Gain (Vout/Vin)
Low-Pass	Active		$1 + \frac{R_A}{R_B}$
High-Pass	Active		

Where

Transfer function of 2nd order low pass filter

$$H(s) = \frac{{\omega_C}^2}{s^2 + 2\omega_C s + {\omega_C}^2} \quad (37)$$

Transfer function of 2nd order high pass filter

$$H(s) = \frac{s^2}{s^2 + 2\omega_C s + {\omega_C}^2} \quad (38)$$

Cutoff frequency (same capacitor and resistor values)

$$f_C = \frac{1}{2\pi RC} \quad (39)$$

$$\omega_C = 2\pi f_C \quad (40)$$

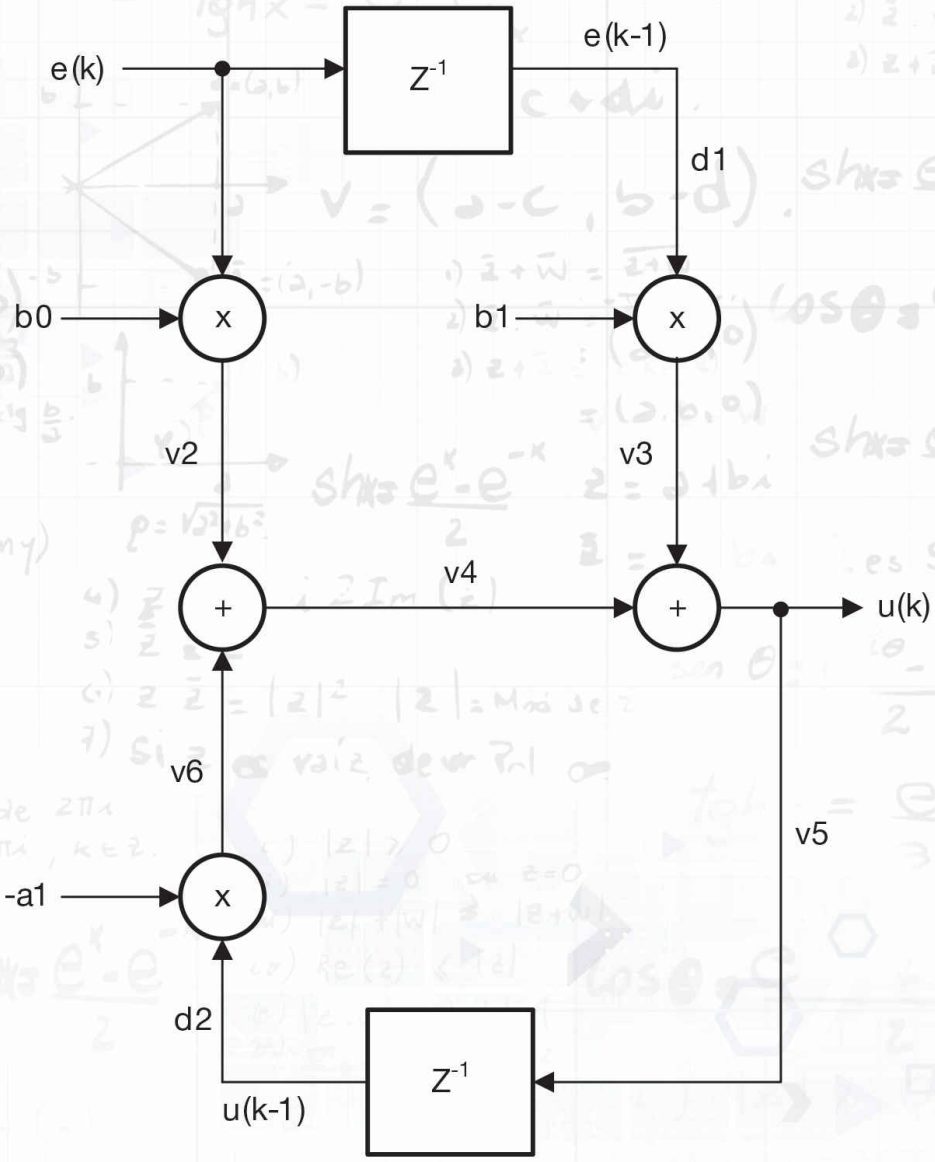
Cutoff frequency (different capacitor and resistor values)

$$f_C = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}} \quad (41)$$

$$\omega_C = 2\pi f_C \quad (42)$$

Notes

Controllers



Linear PID

Linear Proportional Integral Derivative (PID) controllers calculate an error value, e(t), based on the set point and the process variable, and apply a correction based on a proportional, integral, and derivative term. A PID controller is a particular case of a **2P2Z** controller, where A1 = -1 and A2 = 0.

Term	Description	Component
Proportional	The proportional component depends only on the difference between the setpoint and the process variable	$u(t) = K_p e(t)$
Integral	The integral component sums the error term over time, giving a complete controller error history up to the present time. This effectively causes an increase over time unless the error is zero, so the effect is to drive the steady-state error to zero.	$u(t) = K_i \int_0^t e(\tau) d\tau$
Derivative	The differential component causes the output to decrease if the process variable is increasing rapidly. The derivative response is proportional to the rate of change of the process variable.	$u(t) = K_d \frac{de(t)}{dt}$

Proportional:

Where

$e(t)$ = error value

K_p = proportional gain

K_i = integral gain

K_d = derivative gain

Parallel Linear PID

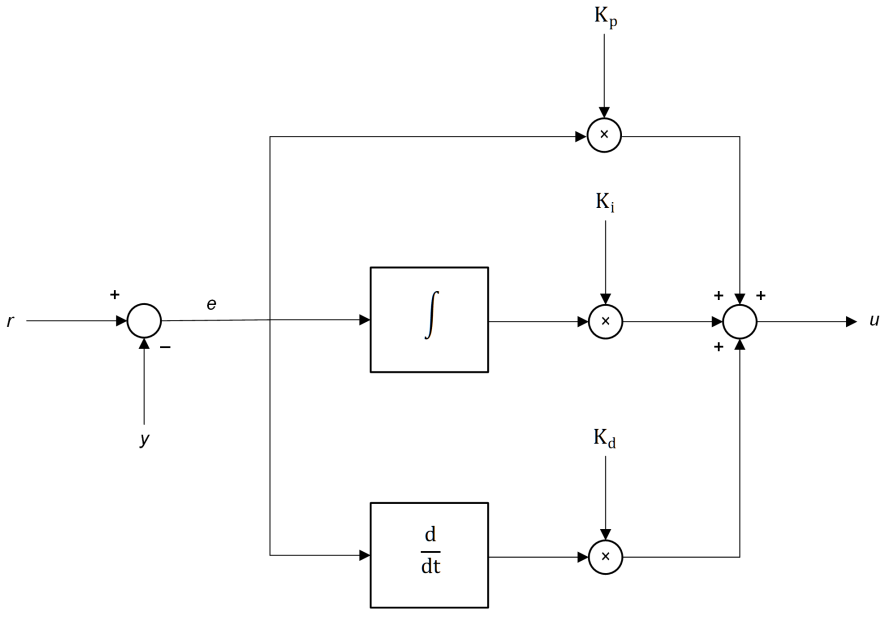


Figure 8. *Parallel form linear PID controller.*

Where

r = input

y = feedback

e = error

u = output

K_p = proportional gain

K_i = integral gain

K_d = derivative gain

Series Linear PID

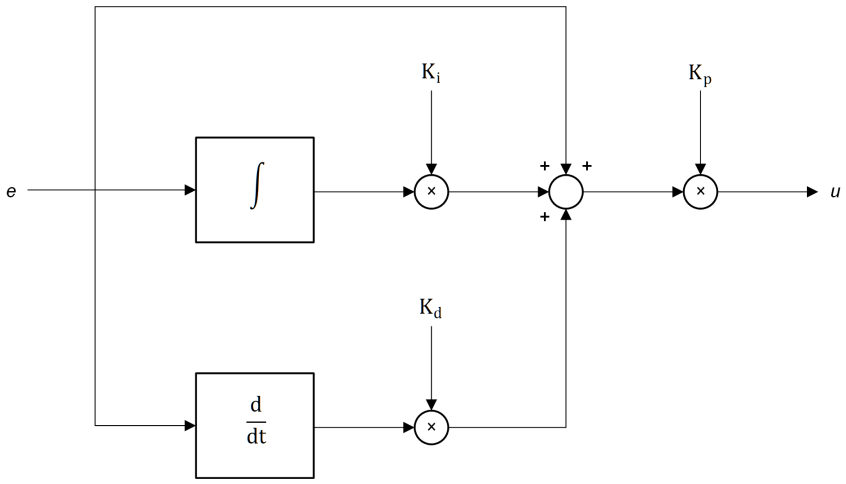


Figure 9. *Series form of linear PID controller.*

Where

e = error

u = output

K_p = proportional gain

K_i = integral gain

K_d = derivative gain

PID equation

Control law for PID

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (43)$$

Where

K_p = proportional gain

K_i = integral gain

K_d = derivative gain

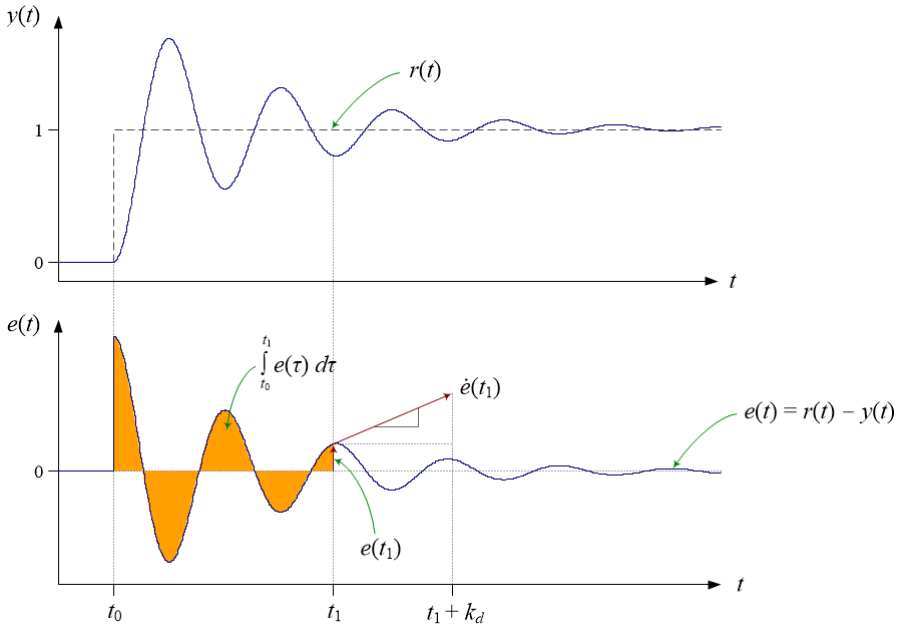


Figure 10. *PID control action.*

Linear PI

Linear proportional integral controllers are like linear PID controllers, but only have two tuning parameters, a proportional term and an integral term.

Equation

Control law for linear PI

$$u(t) = u_{bias} + K_p e(t) + K_i \int_0^t e(t)dt \tag{44}$$

Where

u_{bias} = controller bias set by the error when the controller is switched from manual to automatic mode

K_p = proportional gain

K_i = integral gain

$e(t)$ = error value

Nonlinear PID

A nonlinear proportional integral derivative (Nonlinear PID) controller utilizes a power function to implement the control law. The NLPID is an adaptation of the linear PID in which a non-linear law based on a power function is placed in series with each path.

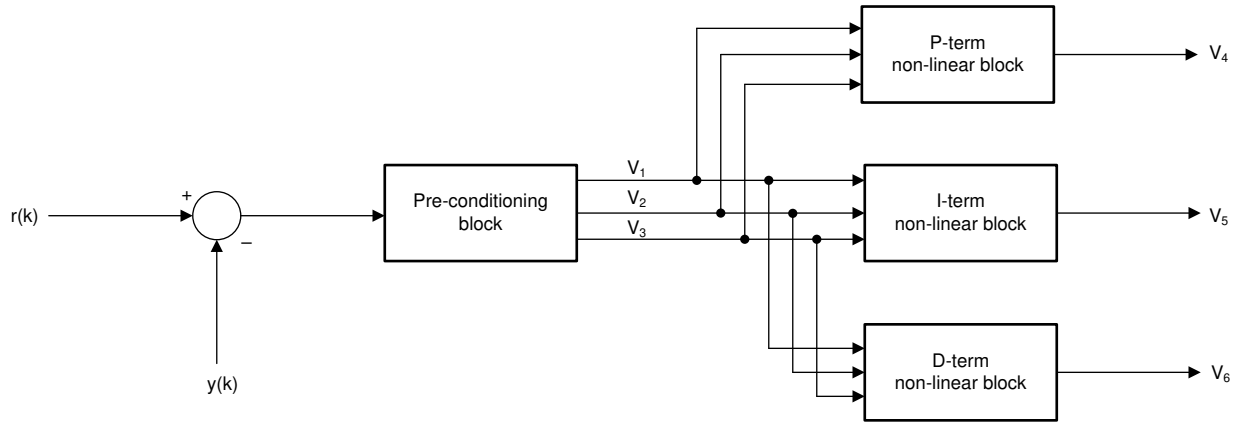


Figure 11. *Nonlinear PID input architecture.*

Tuning parameter

Each nonlinear block shapes the servo error according to a power function law in which the normalized input (the servo error) is raised to the power of an adjustable tuning parameter, α . The tuning parameter determines the degree and direction of the gain shape.

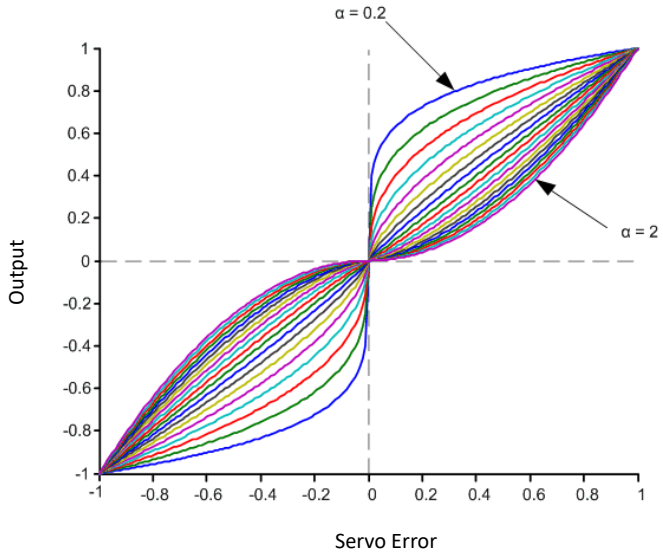


Figure 12. *Nonlinear control law input-output plot.*

Table 7. *Tuning parameter effect on the gain shape.*

Parameter	Value	Effect
α_p	< 1	Smaller gain as error is large, not sensitive to small error.
α_p	> 1	Higher gain when error is large, higher gain when error is small and by that more sensitive to small changes.
α_i	$-1 < \alpha_i < 0$	Solves integral windup problem by reducing the integral action when error is large.
α_d	$\alpha_d > 1$	Makes the differential gain small when the error is small which results in less sensitivity to noise.

To prevent undesired results, the solution is to define an input range covering the origin over which the gain is held constant. The gain in this region is chosen to ensure that linear and nonlinear curves intersect precisely at their boundaries, resulting in a smooth, glitch-free transition from one region to the other.

Equations

Nonlinear control law

$$y(x, \alpha, \delta) = \begin{cases} |x|^{\alpha} \text{sign}(x), & \text{when } |x| \geq \delta \\ \delta^{\alpha - 1} x, & \text{when } |x| < \delta \end{cases} \quad (45)$$

Proportional error expression

$$e_p = e \quad (46)$$

Integral error expression

$$e_i = \int e dt \quad (47)$$

Derivative error expression

$$e_d = \frac{d}{dt} e \quad (48)$$

Reconstructed nonlinear control law

$$u = K_p y(e_p, \alpha_p, \delta_p) + K_i y(e_i, \alpha_i, \delta_i) + K_d y(e_d, \alpha_d, \delta_d) \quad (49)$$

Where

x = input

y = output

α = tuning parameter

δ = logarithmic decrement

K = controller gain

Figure 13 shows the linear and nonlinear regions for a tuning parameter less than one. Notice that the linear gain is independent of the input x, so it does not need to be computed each time the controller runs. The linear gain is fixed for each path and needs to only be recomputed when either of the nonlinear parameters in that path is adjusted.

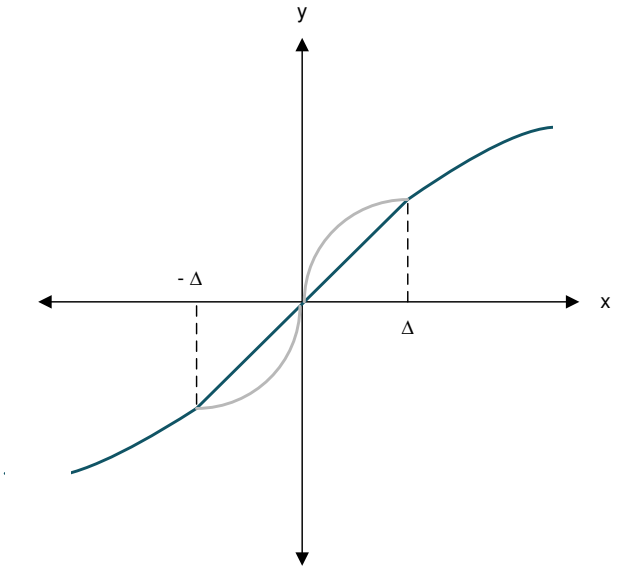


Figure 13. *Nonlinear PID linearized region.*

2P2Z

The 2-Pole/2-Zero (2P2Z) compensator is a filter which introduces a specific gain and phase boost into the system considering two poles and two zeros.

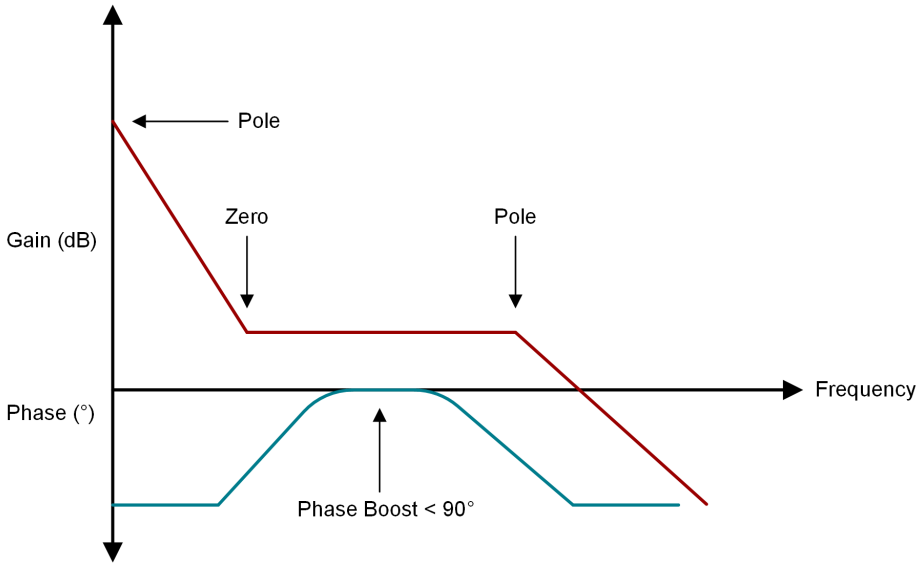


Figure 14. *Phase and gain characteristics of a 2P2Z compensator.*

Equation

Transfer function of 2P2Z compensator

$$H_c(s) = K_{DC}\frac{\omega_{p0}}{s}\frac{\left(1+\frac{s}{w_{z1}}\right)}{\left(1+\frac{s}{w_{P2}}\right)} \quad (50)$$

3P3Z

The 3-Pole/3-Zero (3P3Z) compensator is a filter that introduces a specific gain and phase boost into the system considering three poles and three zeros.

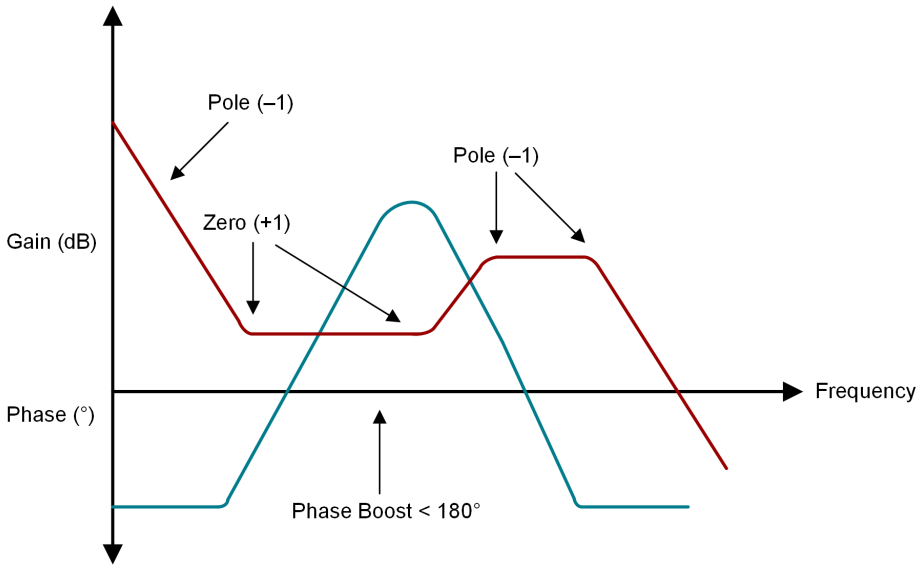


Figure 15. *Phase and gain characteristics of a 3P3Z compensator.*

Equation

Transfer function of 3P3Z compensator

$$H_c(s) = K_{DC}\frac{\omega_{p0}}{s}\frac{\left(1+\frac{s}{w_{z1}}\right)\left(1+\frac{s}{w_{z2}}\right)}{\left(1+\frac{s}{w_{P2}}\right)\left(1+\frac{s}{w_{P3}}\right)} \quad (51)$$

Direct form controllers

DF11

The first order direct form 1 (DF11) compensator implements a first order, or “simple lag” type frequency response.

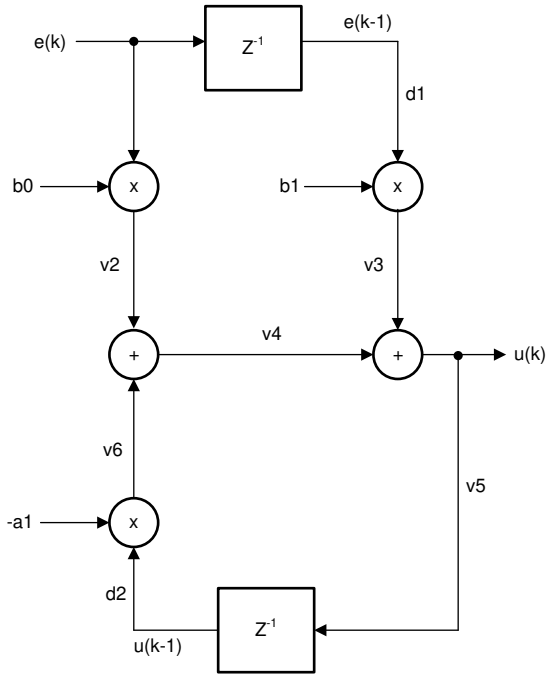


Figure 16. *Representation of DF11.*

Equations

General form of discrete time first order transfer function

$$F(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}} \tag{52}$$

Difference equation

$$u(k) = b_0 e(k) + b_1 e(k-1) - a_1 u(k-1) \tag{53}$$

DF13

The third order direct form 1 (DF13) compensator is a common type of discrete control structure used to implement a control law of dynamical system model specified either as a pole-zero set, or as a rational polynomial in z (a discrete time transfer function). The DF13 controller uses two, three-element delay lines to store previous input and output data required to compute $u(k)$.

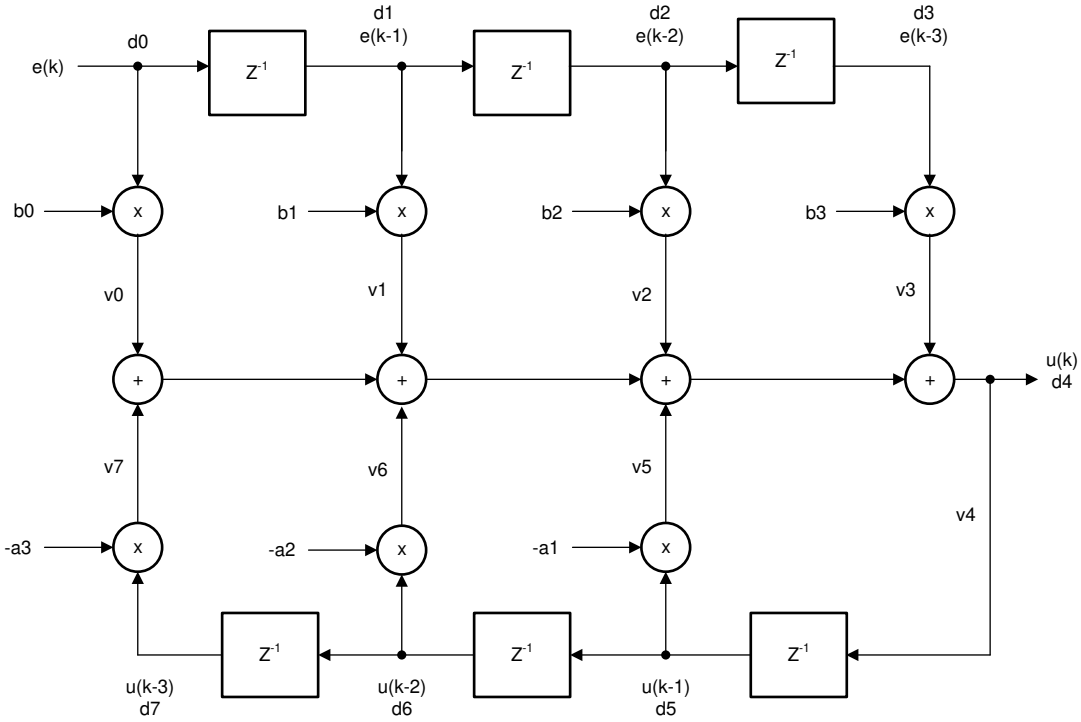


Figure 17. *Representation of DF13.*

Equations

General form of a third order transfer function

$$F(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}} \tag{54}$$

Difference equation

$$u(k) = b_0 e(k) + b_1 e(k-1) + b_2 e(k-2) + b_3 (k-3) - a_1 u(k-1) - a_2 u(k-2) - a_3 u(k-3) \tag{55}$$

DF22

The second order direct form 2 (DF22) compensator is sometimes referred to as a “bi-quad” filter, and is commonly used in a cascaded chain to build up digital filters of higher order.

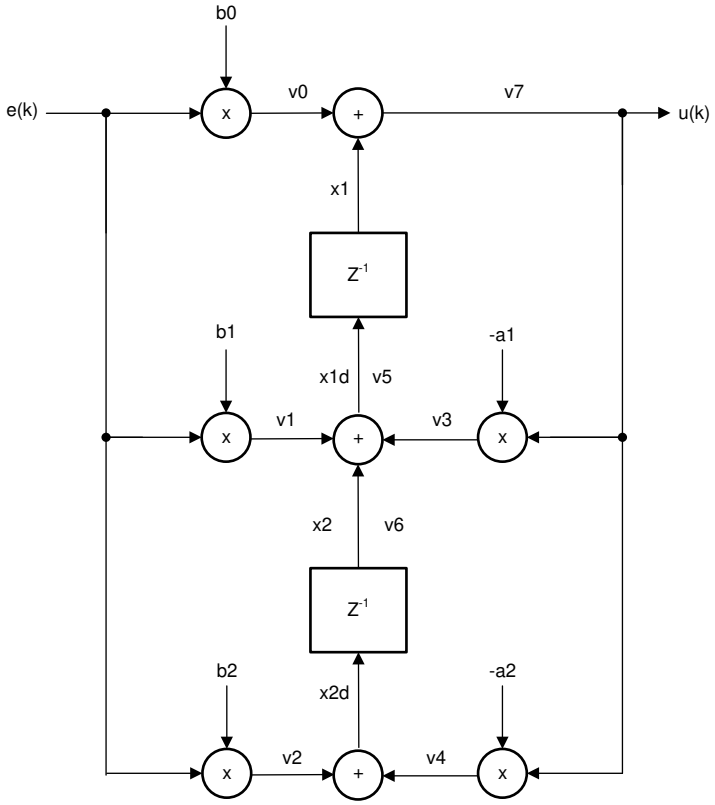


Figure 18. *Representation of DF22.*

Equations

Transfer function of second order discrete time compensator

$$F(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{56}$$

Difference Equation

$$u(k) = b_0 e(k) + b_1 e(k-1) + b_2 e(k-2) - a_1 u(k-1) - a_2 u(k-2) \tag{57}$$

DF23

The third order direct form 2 (DF23) compensator is similar in all respects to the DF22 compensator.

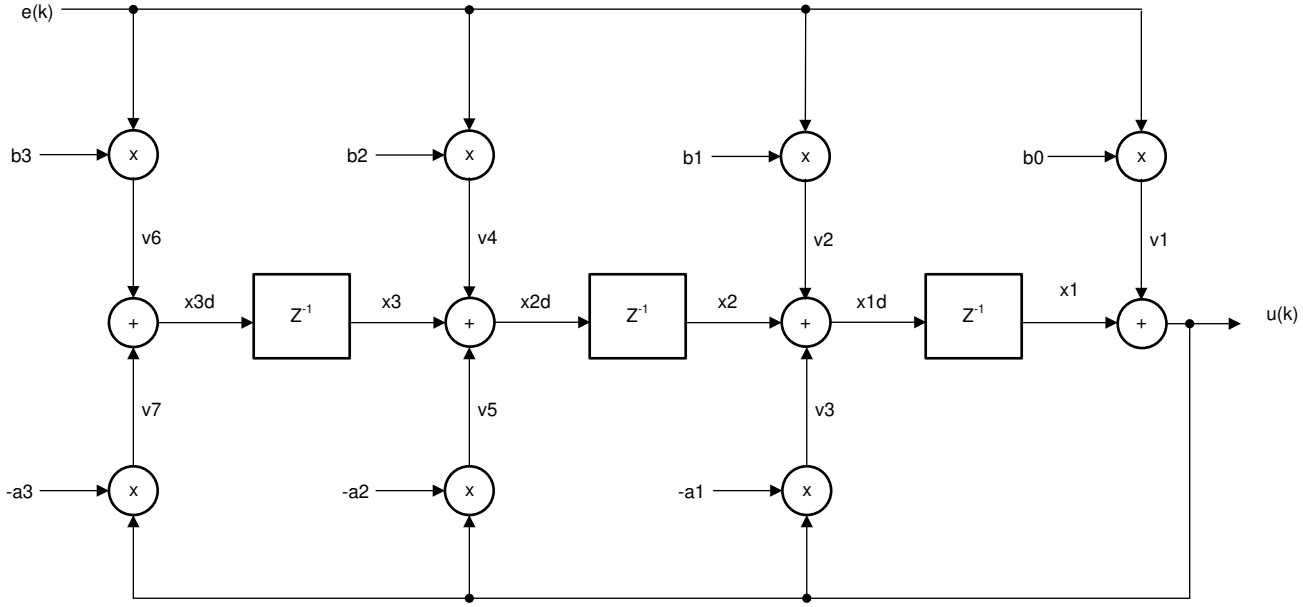


Figure 19. *Representation of DF23.*

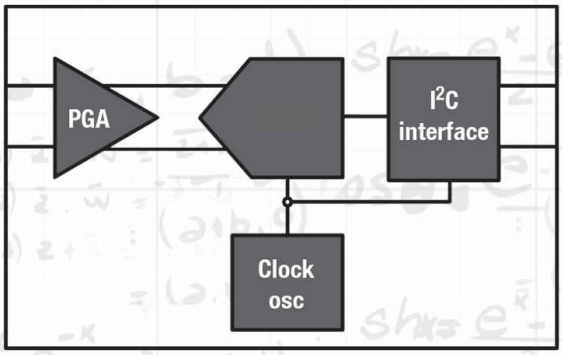
Equation

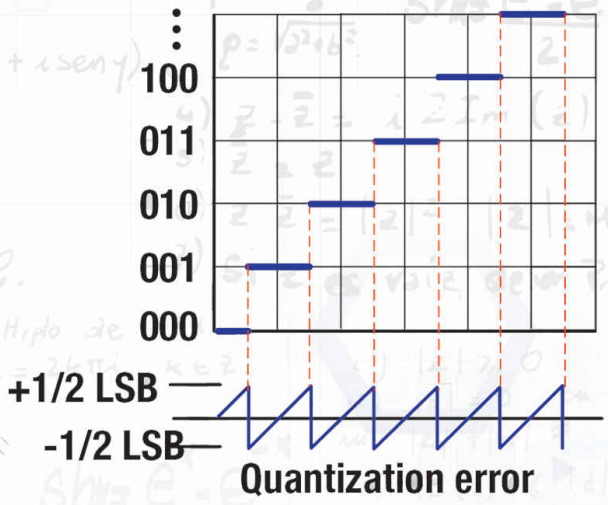
Control law equation

$$u(k) = b_0 e(k) + b_1 e(k-1) + b_2 e(k-2) + b_3 (k-3) - a_1 u(k-1) - a_2 u(k-2) - a_3 u(k-3) \quad (58)$$

Notes

ADC





ADC definitions

Resolution = n	The number of bits used to quantify the input
Number of codes = 2^n	The number of output code combinations
Full-scale range input = FSR	Sets the converter input range and the LSB voltage
LSB = FSR / 2^n	The voltage step size of each LSB
Full-scale input voltage = (2^n – 1) • 1LSB	Full-scale input voltage of the analog-to-digital converter (ADC)
Full-scale output code = 2^n – 1	Largest code that can be read
Transfer function: Output Code = FLOOR [VIN / (FSR/2^n)]	Relationship between input voltage and output code

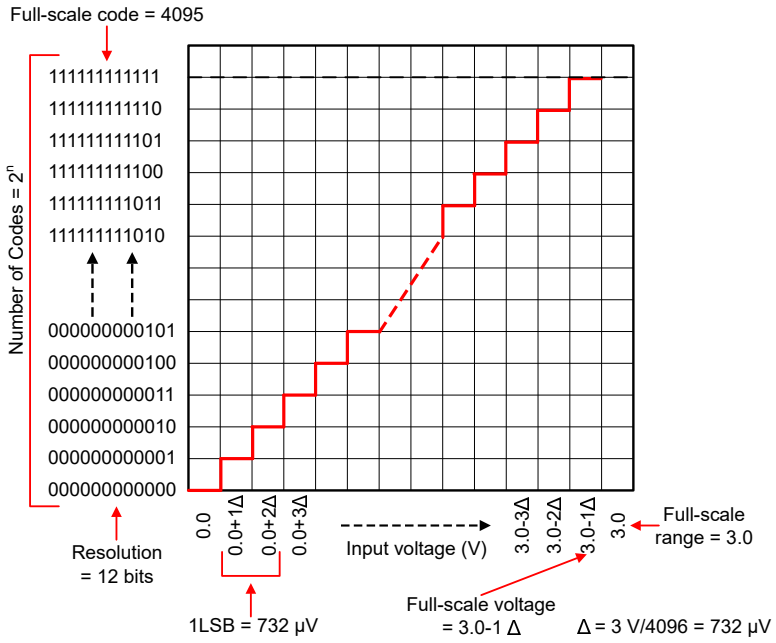


Figure 20. *ADC transfer function.*

ADC resolution

ADC resolution for unipolar

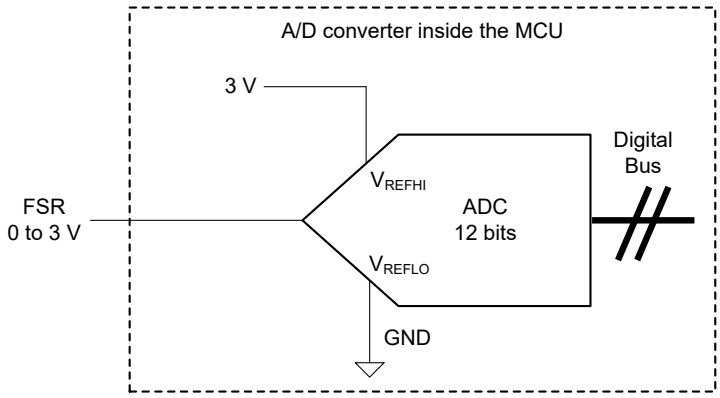


Figure 21. *ADC full-scale range (FSR) unipolar.*

Full-scale range (FSR) unipolar equations

Full Scale Range

$$FSR = V_{REFHI} - V_{REFLO} \quad (59)$$

One Least Significant Bit

$$1LSB = \frac{FSR}{2^n} \quad (60)$$

Where

FSR = full-scale range

PGA = PGA gain

1LSB = one least significant bit or resolution of the data converter

n = resolution number of bits

VREF = reference voltage

Example calculation for the circuit above

$$1LSB = \frac{FSR}{2^N} = \frac{2.5\ V}{2^{12}} = 610.35\ \mu V$$

ADC resolution for differential signals

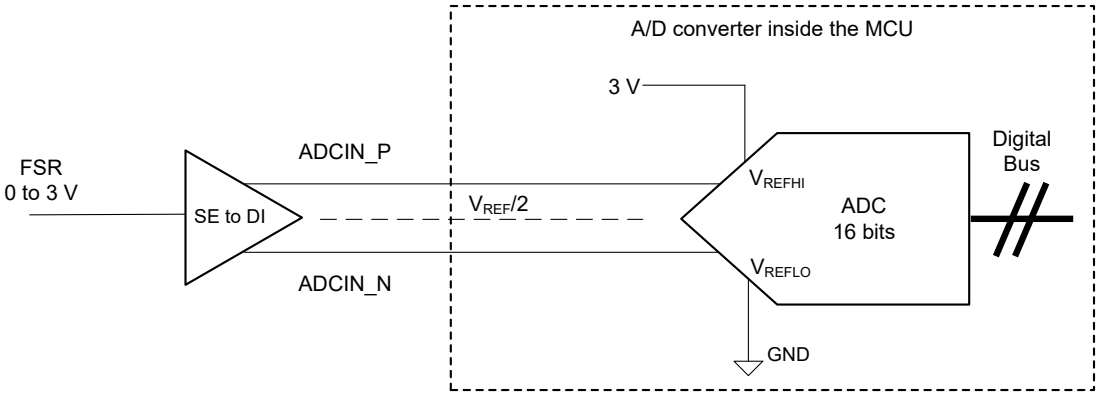


Figure 22. *ADC full-scale range (FSR) differential.*

Where

FSR = full-scale range

1LSB = one least significant bit or resolution of the data converter

n = resolution number of bits

V_{REFHI} = reference voltage high

V_{REFLO} = reference voltage low

Resolution voltage vs. full-scale range

Table 8. *LSB voltage vs. resolution and reference voltage.*

Resolution	FSR (Full-Scale Range)		
	2.5 V	**3V**	**3.3V**
8	9.67 mV	11.7 mV	12.9 mV
10	2.44 mV	2.93 mV	3.222 mV
12	610 µV	732 µV	806 µV
14	152.5 µV	183 µV	201 µV
16	38.14 µV	45.77 µV	50.35 µV
18	9.53 µV	11.44 µV	12.58 µV
20	2.384 µV	2.861 µV	3.147 µV
22	596 nV	715 nV	787 nV
24	149 nV	179 nV	196 nV

Quantization error of ADC

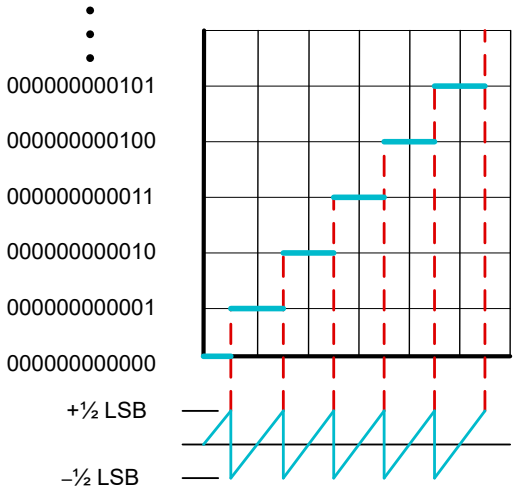


Figure 23. *Quantization error of an ADC converter.*

Quantization error

The error introduced as a result of the quantization process. The amount of this error is a function of the resolution of the converter. The quantization error of an ADC converter is ½ LSB. The quantization error signal is the difference between the actual voltage applied and the ADC output (**Figure 23**). The RMS of the quantization signal is $\mathrm{RMSNoise} = \frac{1\ \mathrm{LSB}}{\sqrt{12}}$.

Signal-to-noise ratio (SNR) from quantization noise only

$$\mathrm{MaxRMSSignal} = \frac{\mathrm{FSR}/2}{\sqrt{2}} = \frac{1\ \mathrm{LSB} \times 2^{\mathrm{N}-1}}{\sqrt{2}} \tag{61}$$

$$\mathrm{RMSNoise} = \frac{1\ \mathrm{LSB}}{\sqrt{12}} \tag{62}$$

$$\mathrm{SNR} = \frac{\mathrm{MaxRMSSignal}}{\mathrm{RMSNoise}} = \frac{1\ \mathrm{LSB} \times 2^{\mathrm{N}-1}/\sqrt{2}}{1\ \mathrm{LSB}/\sqrt{12}} = 2^{\mathrm{N}-1}\sqrt{6} \tag{63}$$

$$\mathrm{SNR(dB)} = 20\log(\mathrm{SNR}) = [20\log(2)]\mathrm{N} + 20\log\left(\frac{\sqrt{6}}{2}\right) \quad (64)$$

$$\mathrm{SNR(dB)} \approx 6.02\mathrm{N} + 1.76 \quad (65)$$

Where

FSR = full-scale range of the ADC converter

1LSB = the voltage of 1LSB,VREF/2^n

N = the resolution of the ADC converter

MaxRMSSignal = the RMS equivalent of the ADC's full-scale input

RMSNoise = the RMS noise from quantization

SNR = the ratio of RMS signal to RMS noise

Example

What is the SNR for an 8-bit ADC converter with 5 V reference, assuming only quantization noise?

Answer

$$SNR = 2^{N-1}\sqrt{6} = 2^{8-1}\sqrt{6} = 314 \quad (66)$$

SNR(dB) = 20log(314)

SNR(dB) = 6.02(8) + 1.76 = 49.9 dB

Total harmonic distortion (THD)

Total harmonic distortion (VRMS)

$$\mathrm{THD(\%)} = \left(\frac{\mathrm{RMSDistortion}}{\mathrm{MaxRMSSignal}}\right) \times 100 = \frac{\sqrt{\mathrm{V_2^2} + \mathrm{V_3^2} + \mathrm{V_4^2} + \ldots + \mathrm{V_n^2}}}{\mathrm{V_1}} \times 100 \quad (67)$$

$$\mathrm{THD(dB)} = 20\log\left(\frac{\mathrm{RMSDistortion}}{\mathrm{MaxRMSSignal}}\right) \quad (68)$$

Where

THD = Total Harmonic Distortion, the ratio of the RMS distortion to the RMS signal

RMSDistortion = the RMS sum of all harmonic components

MaxRMSSignal = the RMS value of the input signal

V1 = the fundamental, generally the input signal

V2, V3, V4, …Vn = harmonics of the fundamental

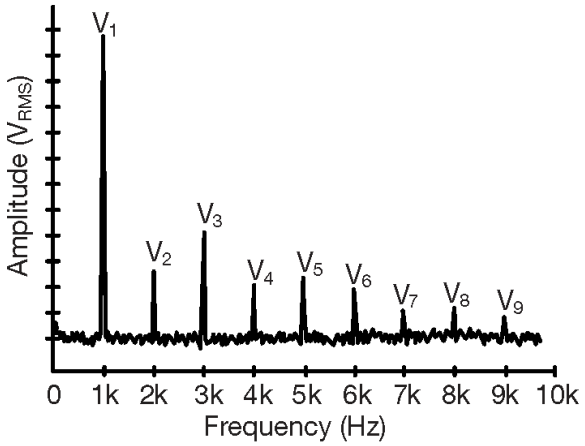


Figure 24. *Fundamental and harmonics in VRMS.*

Total harmonic distortion (dBc)

$$\mathrm{THD(dBc)} = 10\log\left[10\left(\frac{D_2}{10}\right) + 10\left(\frac{D_3}{10}\right) + 10\left(\frac{D_4}{10}\right) + \ldots + 10\left(\frac{D_n}{10}\right)\right] \quad (69)$$

Where

THD = Total Harmonic Distortion. The ratio of the RMS distortion to the RMS signal.

D1 = the fundamental, generally the input signal. This is normalized to 0 dBc.

D2, D3, D4, …Dn = harmonics of the fundamental measured relative to the fundamental

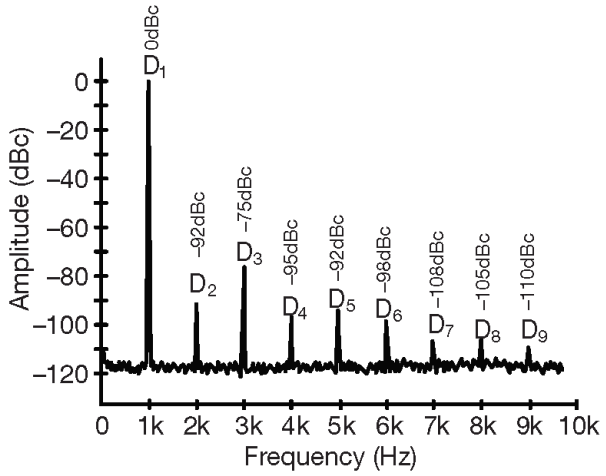


Figure 25. *Fundamental and harmonics in dBc.*

Example

Determine THD for the example above.

Answer

$$\text{THD (dBc)} = 10\log\left[10^{\left(\frac{-92}{10}\right)} + 10^{\left(\frac{-75}{10}\right)} + 10^{\left(\frac{-95}{10}\right)} + \ldots + 10^{\left(\frac{-110}{10}\right)}\right]$$

$$\text{THD(dBc)} = -74.76\ \text{dB}$$

AC signals

Signal-to-Noise and Distortion (SINAD)

$$\text{SINAD(dB)} = 20\log\left(\frac{\text{MaxRMSSignal}}{\sqrt{\text{RMSNoise}^2 + \text{RMSDistortion}^2}}\right) \quad (70)$$

$$\text{SINAD(dB)} = -20\log\left(\sqrt{10^{\left(\frac{-\text{SNR(dB)}}{10}\right)} + 10^{\left(\frac{\text{THD(dB)}}{10}\right)}}\right) \quad (71)$$

Effective Number of Bits (ENOB)

$$\text{ENOB} = \frac{\text{SINAD(dB)} - 1.76\text{dB}}{6.02} \quad (72)$$

Where

MaxRMSSignal = the RMS equivalent of the ADC's full-scale input

RMSNoise = the RMS noise integrated across the ADC converters

RMSDistortion = the RMS sum of all harmonic components

SINAD = the ratio of the full-scale signal-to-noise ratio and distortion

THD = Total harmonic distortion. The ratio of the RMS distortion to the RMS signal.

SNR = the ratio of RMS signal to RMS noise

Example

Calculate the SNR, THD, SINAD and ENOB given the following information:

MaxRMSSignal = 1.76V_{RMS}

VRMS RMSDistortion = 50 μV_{RMS}

RMSNoise = 100 μV_{RMS}

Answer

$$\text{SNR(dB)} = 20\log\left(\frac{1.76\ \text{V}_{\text{RMS}}}{100\ \mu\text{V}_{\text{RMS}}}\right) = 84.9\ \text{dB}$$

$$\text{THD(dB)} = 20\log\left(\frac{50\ \mu\text{V}_{\text{RMS}}}{100\ \mu\text{V}_{\text{RMS}}}\right) = -90.9\ \text{dB}$$

$$\text{SINAD(dB)} = 20\log\left(\frac{1.76\ \text{V}_{\text{RMS}}}{\sqrt{(100\ \mu\text{V}_{\text{RMS}})^2 + (50\ \mu\text{V}_{\text{RMS}})^2}}\right) = 83.9\ \text{dB}$$

$$\text{SINAD(dB)} = -20\log\left(\sqrt{10^{\left(\frac{-84.9\ \text{dB}}{10}\right)} + 10^{\left(\frac{-90.9\ \text{dB}}{10}\right)}}\right) = 83.9\ \text{dB}$$

$$\text{ENOB} = \frac{83.9\ \text{dB} - 1.76\ \text{dB}}{6.02} = 13.65$$

DC signals

$$\text{NoiseFreeResolution} = \log_2\left(\frac{2^N}{\text{PeaktoPeakNoiseinLSB}}\right) \quad (73)$$

$$\text{EffectiveResolution} = \log_2\left(\frac{2^N}{\text{rmsNoiseLSB}}\right) \quad (74)$$

$$\text{PeaktoPeakNoiseinLSB} \approx 6.6 \times \text{rmsNoiseLSB} \quad (75)$$

$$\text{EffectiveResolution} \approx \text{NoiseFreeResolution} + 2.7 \quad (76)$$

Note

The maximum effective resolution is never greater than the ADC resolution. For example, a 24-bit converter cannot have an effective resolution greater than 24 bits.

Example

What is the noise-free resolution and effective resolution for a 24-bit converter assuming the peak-to-peak noise is 7 LSBs?

Answer

$$\text{NoiseFreeResolution} = \log_2\left(\frac{2^{24}}{7}\right) = 21.2$$

$$\text{EffectiveResolution} = \log_2\left(\frac{2^{24}}{\frac{7}{6.6}}\right) = 23.9$$

$$\text{EffectiveResolution} = 21.2 + 2.7 = 23.9$$

Settling time and conversion accuracy

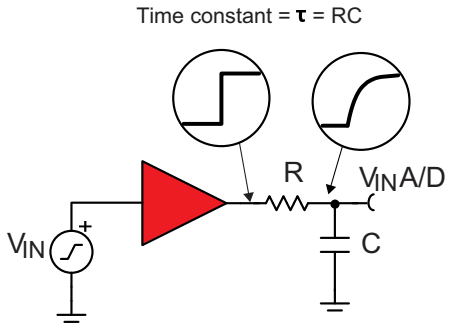


Figure 26. *Settling time and conversion accuracy.*

Table 9. *Conversion accuracy achieved after a specified time.*

Settling Time in Time Constants (N_{TC})	Accuracy in Bits (N)	Settling Time in Time Constants (N_{TC})	Accuracy in Bits
1	1.44	10	14.43
2	2.89	11	15.87
3	4.33	12	17.31
4	5.77	13	18.76
5	7.21	14	20.20
6	8.66	15	21.64
7	10.10	16	23.08
8	11.54	17	24.53
9	12.98	18	25.97

Where

N = the number of bits of accuracy the RC circuit has settled to after N_{TC} number of time constants

N_{TC} = the number of RC time constants. Where one time constant equals R·C.

Note

For a FSR step. For single-ended input ADC with no PGA front end, FSR (full-scale range) = V_{REF}.

Table 10. *Time required to settle to a specified conversion accuracy.*

Settling Time in Time Constants (N_{TC})	Accuracy in Bits (N)	Settling Time in Time Constants (N_{TC})	Accuracy in Bits
8	5.5	17	11.78
9	6.24	18	12.48
10	6.93	19	13.17
11	7.62	20	13.86
12	8.32	21	14.56
13	9.01	22	15.25
14	9.70	23	15.94
15	10.40	24	16.64
16	11.04	25	17.33

Where

N_{TC} = the number of time constants required to achieve N bits of settling. Where one time constant equals R·C.

N = the number of bits of accuracy

Note

For a FSR step. For single-ended input ADC with no PGA front end, FSR (full-scale range) = V_{REF}.

ADC system noise

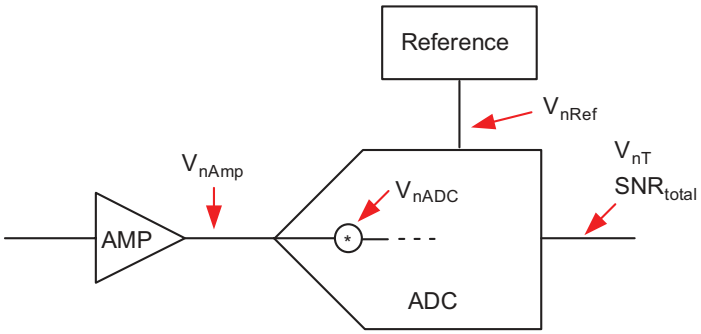


Figure 27. *ADC noise.*

$$V_{FSR\_RMS} = \frac{V_{FSR} \times 0.707}{2} \quad (77)$$

$$SNR_{ADC} = 20 \times \log\left(\frac{V_{FSR\_RMS}}{V_{nADC}}\right) \quad (78)$$

$$V_{nADC} = \frac{V_{FSR\_RMS}}{10\left(\frac{SNR_{ADC}}{20}\right)} \quad (79)$$

$$V_{nT} = \sqrt{(V_{nADC})^2 + (V_{nAmp})^2 + (V_{nRef})^2} \quad (80)$$

Where

VFSR = the ADC full scale range from the data sheet

V_{FSR_RMS} = This finds the maximum RMS amplitude of a sine wave applied to an ADC. Dividing the ADC full scale range by 2 converts peak-to-peak to peak. Multiplying by 0.707 converts to RMS.

SNR_{ADC} = Data converter signal to noise ratio specification from data sheet

V_{nADC} = Noise in volts RMS derived from the SNR equation. Converting noise to volts allows it to be combined with amplifier and reference noise.

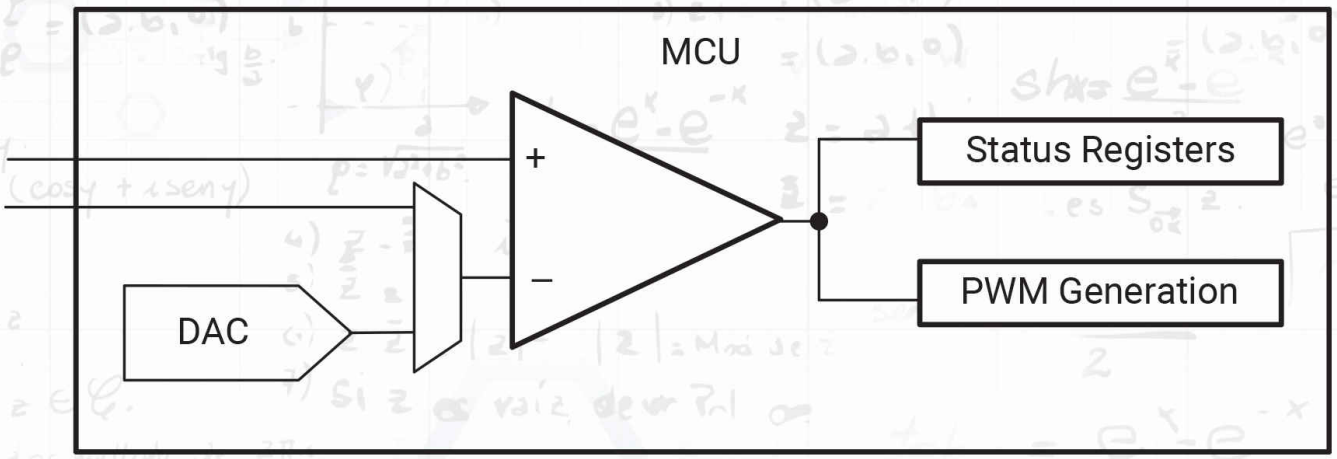
V_{nAmp} = Amplifier noise in volts RMS calculated or simulated using data sheet parameters

V_{nRef} = Reference noise in volts RMS calculated or simulated using data sheet parameters

V_{nT} = Total noise in volts RMS calculated by combining ADC, amplifier, and reference noise

Notes

Comparator



Basic operation

Table 11 gives the expected behavior of the comparator output based on the state of the inputs.

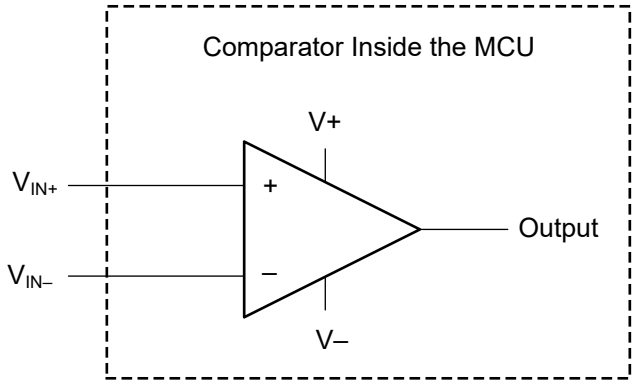


Figure 28. *Comparator diagram.*

Table 11. *Comparator inputs to output.*

Inputs Condition	Output
IN+ > IN-	HIGH (V+)
IN+ = IN-	Indeterminate(chatters - see Hysteresis)
IN+ < IN-	LOW (V-)

Offset and hysteresis

The basic comparator will have an offset error that is the internal offset between the V_{IN}+ and V_{IN}- inputs. This error term needs to be added to the ideal threshold voltage to determine when the comparator output will change vs a change in its inputs.

The basic comparator configuration may also oscillate or produce noisy "chatter" output if the applied differential input voltage is near the comparator's offset voltage. This usually occurs when the input signal is moving very slowly across the switching threshold of the comparator.

This problem can be prevented by the addition of hysteresis or positive feedback. The hysteresis transfer curve is shown in **Figure 29**. This curve is a function of three components: V_{TH}, V_{OFF}, V_{HYST}.

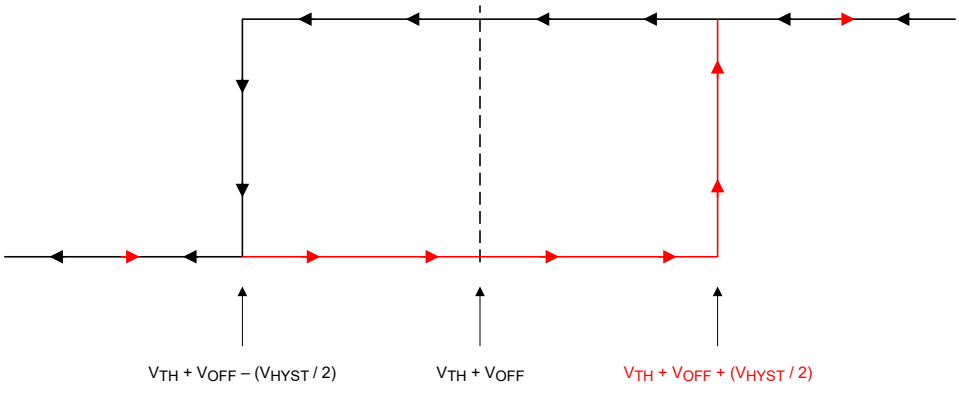


Figure 29. *Comparator hysteresis.*

Where:

- V_{TH} = the actual set voltage or threshold trip voltage
- V_{OFF} = internal offset voltage between V_{IN}+ and V_{IN}-. This voltage is added to V_{TH} to form the actual trip point at which the comparator must respond to change output states
- V_{HYST} = the hysteresis (or trip window) that is designed to reduce comparator sensitivity to noise.

Propagation delay

There is a delay between when the input crosses the reference voltage and the output responds. This is called the propagation delay. Propagation delay can be different based on the rate of change of the inputs. Propagation is most accurately measured by using input signals with a fast ramp such as a digital or square wave.

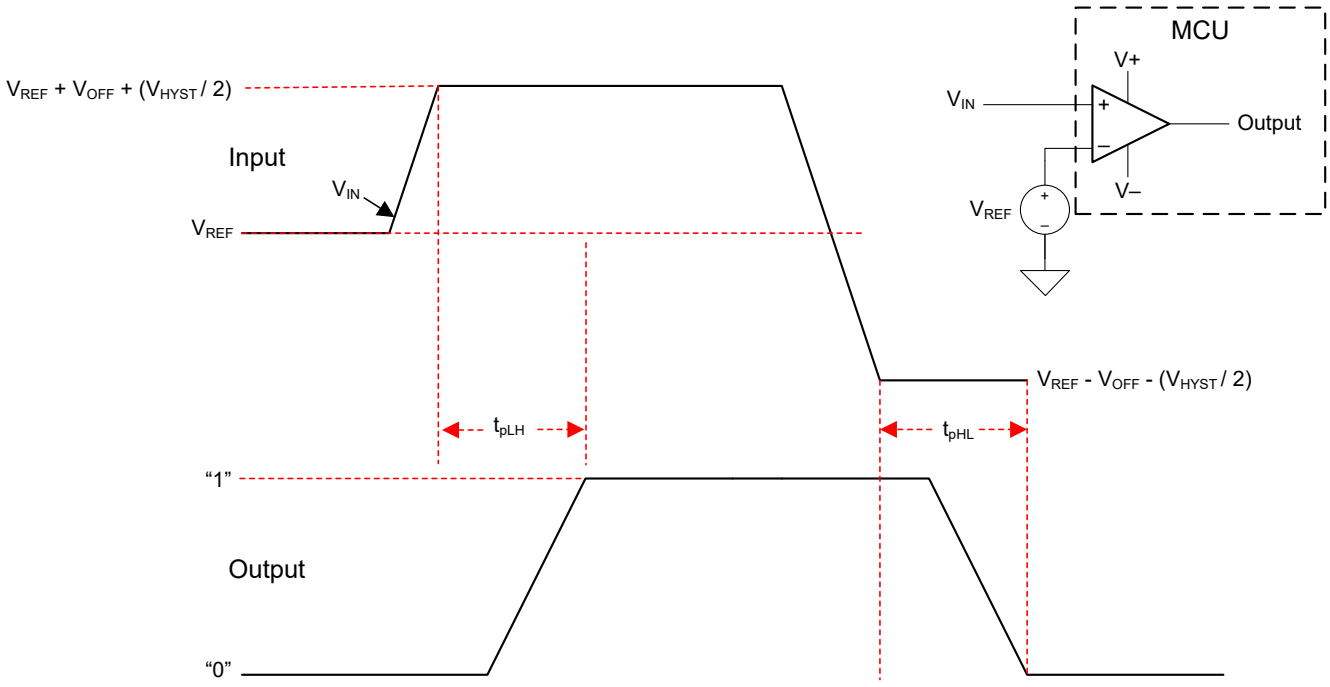
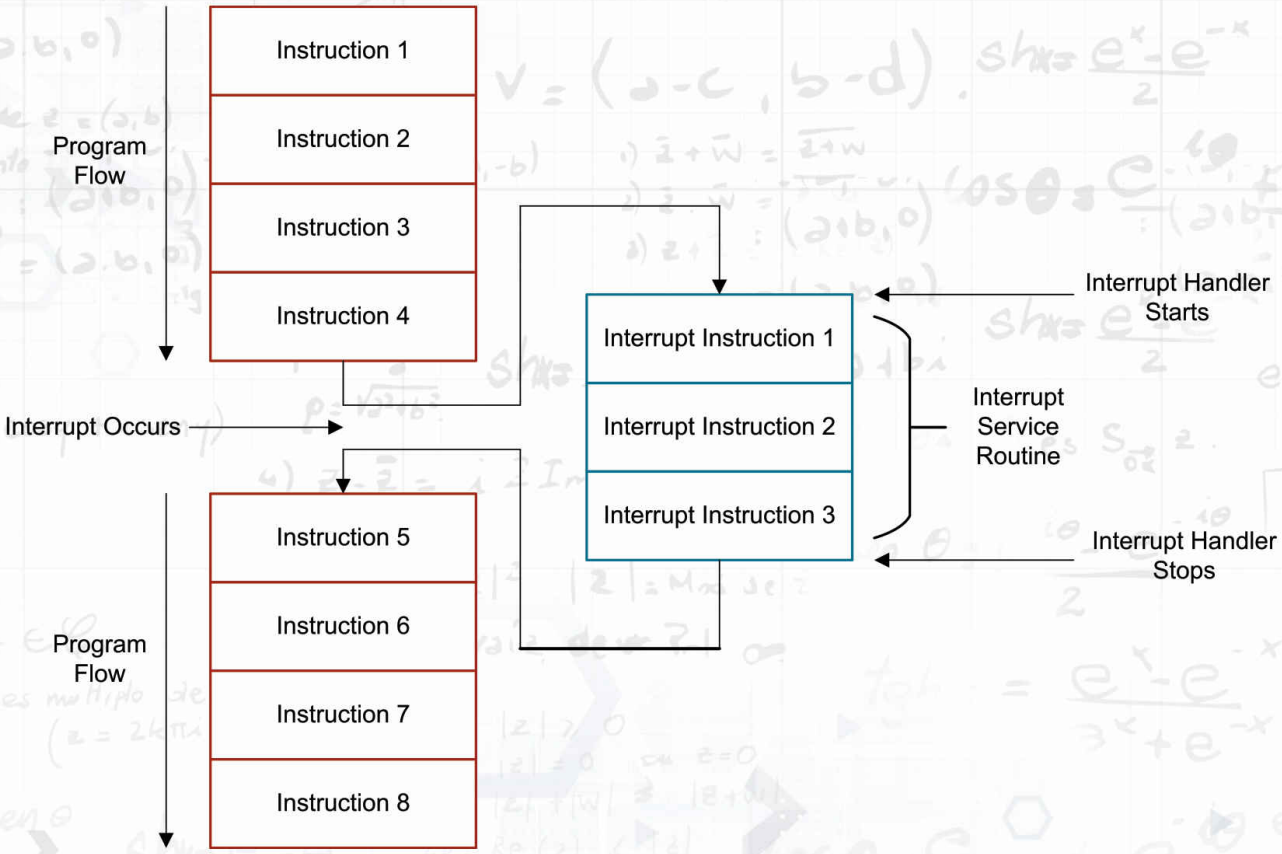


Figure 30. *Comparator input to output switching delay.*

Notes

Processing



Data representation

Table 12. *Primitive data types.*

Type	Default	Size
Boolean	False	1 bit
Byte	0	8 bits
Char	\u000	16 bits
Short	0	16 bits
Int	0	32 bits
Long	0	64 bits
Float	0.0f	32 bits
Double	0.0d	64 bits

Note

Not all CPU architectures use the same bit counts to represent primitive data.

For example, C2000 has a 16-bit architecture and does not support 8-bit types. For a list of data types supported by C2000 devices, see the ***TMS320C28x Optimizing C/C++ Compiler v21.6.0.LTS***.

Note

The size of data pointers depends on the architecture.

Table 13. *Multiplier prefixes and abbreviations.*

Multiplier	Prefix	Abbreviation
10^{9}	Giga	G
10^{6}	Mega	M
10^{3}	Kilo	k
10^{-3}	Milli	m
10^{-6}	Micro	u
10^{-9}	Nano	n
10^{-12}	Pico	p
10^{-15}	Femto	f

Table 14. *ASCII table.*

Binary	Oct	Dec	Hex	Chr
0100000	040	32	20	Space
0100001	041	33	21	!
0100010	042	34	22	"
0100011	043	35	23	#
0100100	044	36	24	$
0100101	045	37	25	%
0100110	046	38	26	&
0100111	047	39	27	'
0101000	050	40	28	(

Binary	Oct	Dec	Hex	Chr
0101001	051	41	29	)
0101010	052	42	2A	*
0101011	053	43	2B	+
0101100	054	44	2C	,
0101101	055	45	2D	-
0101110	056	46	2E	.
0101111	057	47	2F	/
0110000	060	48	30	0
0110001	061	49	31	1
0110010	062	50	32	2
0110011	063	51	33	3
0110100	064	52	34	4
0110101	065	53	35	5
0110110	066	54	36	6
0110111	067	55	37	7
0111000	070	56	38	8
0111001	071	57	39	9
0111010	072	58	3A	:
0111011	073	59	3B	;
0111100	074	60	3C	<
0111101	075	61	3D	=
0111110	076	62	3E	>
0111111	077	63	3F	?
1000000	100	64	40	@
1000001	101	65	41	A
1000010	102	66	42	B
1000011	103	67	43	C
1000100	104	68	44	D
1000101	105	69	45	E
1000110	106	70	46	F
1000111	107	71	47	G
1001000	110	72	48	H
1001001	111	73	49	I
1001010	112	74	4A	J
1001011	113	75	4B	K
1001100	114	76	4C	L
1001101	115	77	4D	M
1001110	116	78	4E	N
1001111	117	79	4F	O
1010000	120	80	50	P
1010001	121	81	51	Q
1010010	122	82	52	R
1010011	123	83	53	S
1010100	124	84	54	T
1010101	125	85	55	U
1010110	126	86	56	V
1010111	127	87	57	W

Binary	Oct	Dec	Hex	Chr
1011000	130	88	58	X
1011001	131	89	59	Y
1011010	132	90	5A	Z
1011011	133	91	5B	[
1011100	134	92	5C	\
1011101	135	93	5D	]
1011110	136	94	5E	^
1011111	137	95	5F	_
1100000	140	96	60	`
1100001	141	97	61	a
1100010	142	98	62	b
1100011	143	99	63	c
1100100	144	100	64	d
1100101	145	101	65	e
1100110	146	102	66	f
1100111	147	103	67	g
1101000	150	104	68	h
1101001	151	105	69	i
1101010	152	106	6A	j
1101011	153	107	6B	k
1101100	154	108	6C	l
1101101	155	109	6D	m
1101110	156	110	6E	n
1101111	157	111	6F	o
1110000	160	112	70	p
1110001	161	113	71	q
1110010	162	114	72	r
1110011	163	115	73	s
1110100	164	116	74	t
1110101	165	117	75	u
1110110	166	118	76	v
1110111	167	119	77	w
1111000	170	120	78	x
1111001	171	121	79	y
1111010	172	122	7A	z
1111011	173	123	7B	{
1111100	174	124	7C	|
1111101	175	125	7D	}
1111110	176	126	7E	~

Central processing unit

CPU basics

A CPU executes all of the instructions within a program. The design of a CPU cannot be altered once it has been manufactured. A CPU's performance is based on many factors and often times requires a benchmark analysis to properly be evaluated. Multiple specifications need to be evaluated together to properly determine a CPU's performance such as the ones defined below.

Definitions:

- **Clock Frequency/Speed:** A measure of the number of cycles the CPU executes per second (usually specified in megahertz (MHz) or gigahertz (GHz)).
- **Cycle:** The amount of time of a singular clock pulse (inverse of the clock speed). Different CPU designs handle instructions differently. Some instructions may require multiple cycles while others may be executed in less than a cycle.
- **MIPS:** Maximum number of instructions that can be executed every cycle.
- **MFLOP:** Maximum number of floating point operations that can be executed every cycle.

CPU pipeline

A CPU pipeline describes the different processing elements that need to be completed either in parallel or sequential order by the different stages of instruction execution. This is embedded in the design of the CPU. During the CPU design, the goal is to optimize the performance of this pipeline by always keeping the different stages busy. The more parallelism there is within the CPU's pipeline the more availability there is to perform multiple instructions at a time. A common set of stages includes:

- Instruction Fetch
- Instruction Decode
- Execution
- Memory Access
- Write Back

Characteristics of a real-time processor

Table 15 contains key characteristics that determine how good a processor is at performing a real-time control task. The term "fast" is used as a relative term to indicate the best performance possible. The execution speed of a complex task is determined by the number of CPU cycles needed to complete the constituent operations.

Table 15. *Real-time control processor - characteristics.*

Characteristic	Description
Fast at performing math operations	Math operations: MUL, ADD, SUB
Fast at trigonometric operations	Trigonometric operations: SIN, COS, ATAN, DIV
Fast at saturation operations	Saturation operations check for out of bounds conditions and if detected the value is clamped or saturated
Contains deterministic execution	A real-time control application has a fixed amount of time to execute algorithms, and if the time taken to perform the algorithms is deterministic (or known) it is easier to budget the available MIPS

Characteristic	Description
Fast access to peripheral registers	For example, reading sensor inputs (example: ADC) and writing to output registers (example: EPWM)
Fast Interrupt Response	The processor needs to respond to periodic interrupt events with low latency to satisfy real time deadlines
Floating-point operation capability	Floating-point operations are generally easier to use which make the control algorithms more robust
32-bit and 64-bit data precision	A processor that can be scaled in precision based on the application is ideal
Multicore Support	Having dual cores or more helps increase the amount of tasks that are executed given a specified amount of time

Signal chain

A signal chain is an important part in evaluating a CPU's performance as it encompasses all of the key components for task operation. Below are the different actions that comprise a signal chain.

- Latch or Respond to Interrupt - The trigger event is latched by hardware which in many real-time control applications triggers ADC sampling upon which interrupt occurs and CPU responds to the interrupt by entering the ISR.
- Context Save - The CPU stores the state of the current process it is doing
- Read Peripheral or Sensor - The CPU has to read the value that has been acquired by a peripheral or sensor*
- Execute the control algorithm - Most of the time within an interrupt will be consumed by the control algorithm since this tends to be the most math intensive portion and requires lots of CPU cycles
- Write the output value - Once the output is calculated it is typically written to a control peripheral*

* = Efficiency in reading or writing to peripherals is a key aspect of a real-time processor. The bus architecture of a device can influence how quickly the CPU can read/write peripherals and can impact processing times as typically control loops will involve reading from and writing to peripherals.

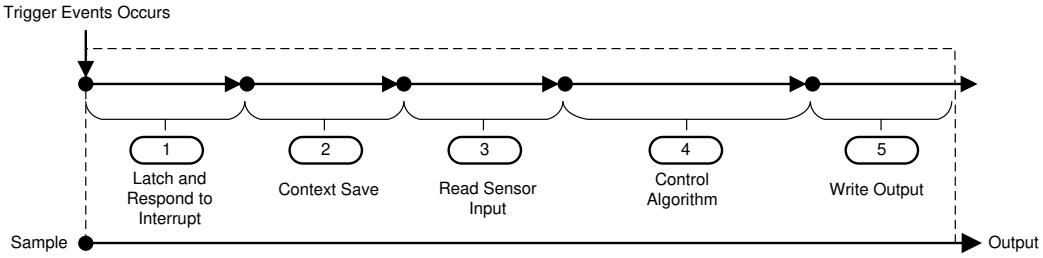


Figure 31. *Signal chain.*

Memory

An important aspect to any real-time control microcontroller is the memory. Various memory types are described within **Table 16**. There are many addressability options as well. Some processors have byte addressable memory while others may have word addressability. Addressability is the smallest unit of content in memory that can be accessed by the CPU. The industry standard for 8-bit addressability is referred to as byte addressable. In contrast, word addressable refers to anything that is not byte addressable. Additionally, some processors might have different word sizes for different tasks.

Table 16. *Memory types.*

Type	Description
FLASH	Flash is a type of nonvolatile memory (NVM). Flash is erased in units called sectors (or blocks) and writes content (data or code) at the word level. A wait-state is a delay experienced by the CPU when accessing a slower memory or interface. Content stored in Flash is not erased after a microcontroller is powered down.
Random Access Memory (RAM)	RAM is used to store data and other results that are created while a microcontroller is at work. Content stored in RAM is erased after the microcontroller is powered down.
Read Only Memory (ROM)	ROM contains specialized tasks that will never change.
CACHE	Cache is smaller, faster memory, located closer to a processor core, which stores copies of the content from frequently used main memory locations. Usually, processors have many levels of cache memory. The cache provides content storage and instructions to prevent the CPU from waiting for content to be retrieved from RAM. Typically, a processesor that runs from cache needs to run at least 2x MHz faster than a processor using a tightly coupled memory system. There can be unpredictable delays when content from cache is refreshed.
Tightly Coupled Memory (TCM)	TCM is memory that is stored on-chip and never cached. This is useful for running code but more importantly to store critical content and as stack space.

Direct memory access (DMA)

Direct Memory Access (DMA) is a feature that allows subsystems or peripherals to access RAM without intervention of the CPU. This is a helpful feature for real-time control applications since it allows the CPU to work on other tasks of the control algorithm, while the DMA transfers data between address locations (both memory and register). The DMA is not always restricted between RAM and other subsystems, but rather can be utilized from memory to memory. A DMA transfer can be started by a peripheral or a software trigger. This capability increases data throughput and allows for a more efficient use of interrupts.

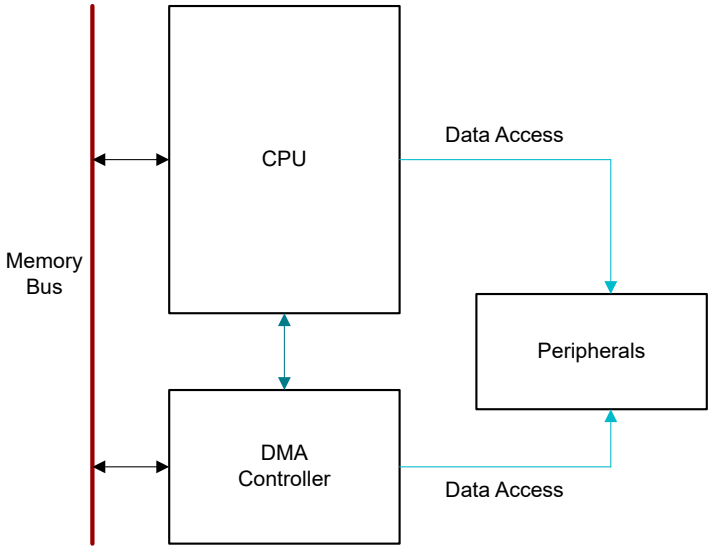


Figure 32. *Direct memory access.*

Interrupts

An interrupt is a response by the processor to an event that needs attention. The processor executes instructions that are defined within the interrupt service routine (ISR) as soon as is capable, and then goes back to normal operational tasks. Unlike idle loops where code waits for an event to occur, interrupts provide the opportunity to have non-idle code loops that execute based on the CPU's availability and priority of tasks.

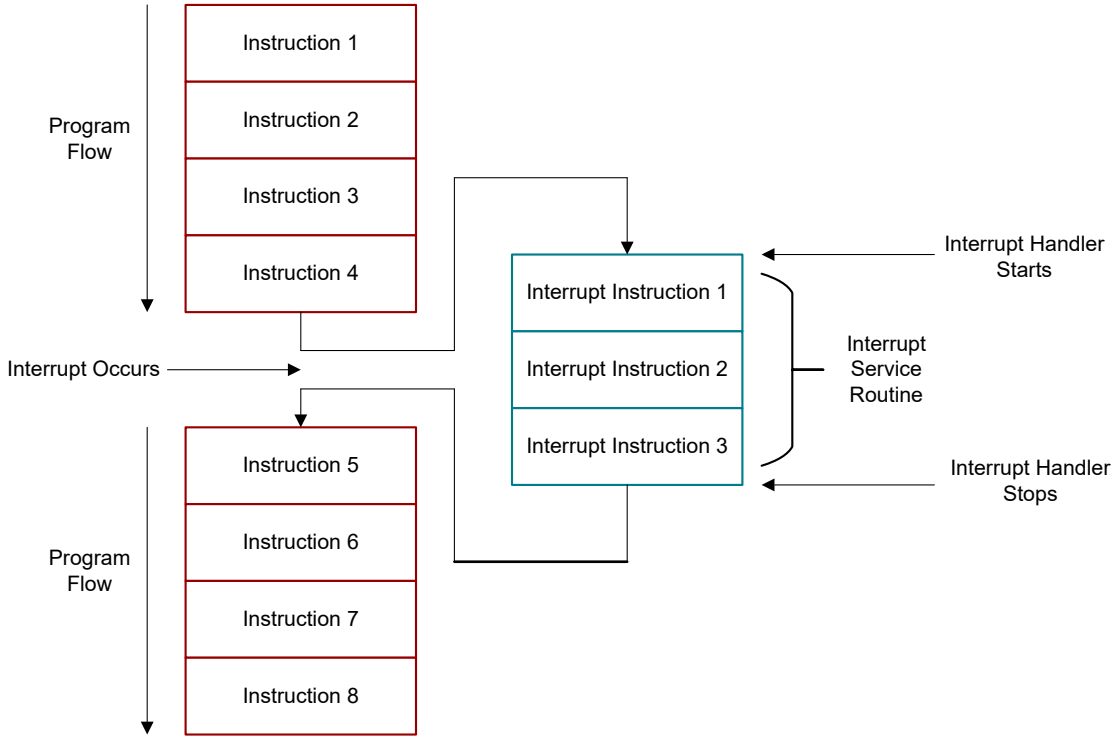


Figure 33. *CPU interrupt flow.*

Interrupt latency is an important factor in understanding the response time of a real-time system. The typical method by which interrupt latency of a system is assessed is the number of cycles it takes for the hardware to respond to an interrupt and branch to the interrupt vector (hardware latch and respond). However, in a real-time application this is only part of the response.

Co-processors and accelerators

Co-Processors

For optimized performance co-processors and accelerators are helpful. A co-processor is used to supplement the functions performed by the main CPU. This creates a model in which multi-threading is utilized. Tasks can be divided across the CPUs depending on the functions to be implemented.

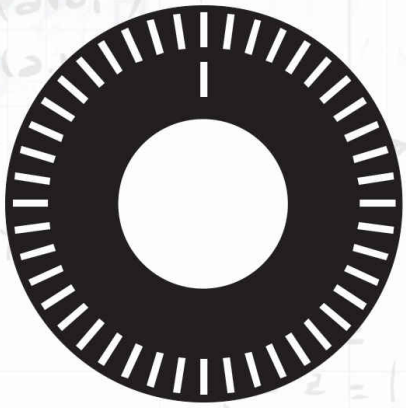
Accelerators

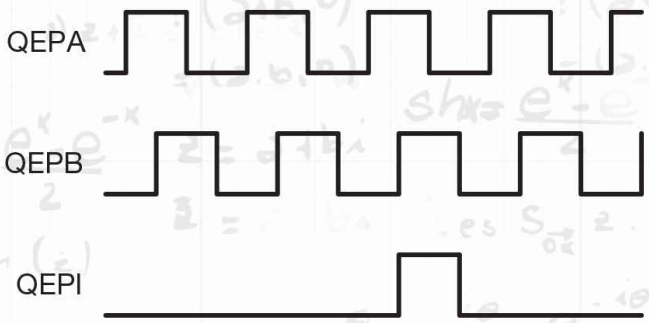
Accelerators are intended to help the CPU perform certain tasks more efficiently by extending the capabilities of a CPU through registers and instructions. The added efficiency can often times remove the need of additional co-processors. Some common accelerators are presented below:

- **Trigonometric Math Unit:** Accelerates several specific trigonometric math operations like sine, cosine, arc tangent, divide, and square root.
- **Cryptographic Unit:** Accelerates encryption algorithms such as the data encryption standard (DES) symmetric encryption algorithm and the advanced encryption system (AES) symmetric encryption algorithm.
- **Floating-Point Unit:** Provides floating-point math support which alleviates scaling and saturation concerns.
- **Complex Math Unit:** Accelerates math operations like add, subtract, and multiply.

Notes

Encoders





Encoder definitions

Resolution	The number of measuring segments or units in one revolution of an encoder shaft or 1 in/mm of a linear scale. Encoder resolution is commonly measured in pulses per revolution (PPR) or lines per revolution (LPR) for incremental encoders. For Absolute encoders it is measured in bits, for example a 12-bit Absolute encoder has resolution of 2^{12}= 4096 counts per revolution.
Speed	Maximum speed at which the device can rotate without sustaining physical damage, often defined as revolutions per minute (RPM).
Positional Accuracy	It is the maximum error of a reading, in arc seconds. One degree is 3600 arc seconds.
Absolute Encoder Codes	Binary, BCD or Gray format.
Incremental Encoder Signal Types	**Quadrature** - Refers to A and B channels that are 90º out of phase. **Single Channel** - Unidirectional allowing one count per physical line. **Pulse & Direction** - Direction channel may be combined with single-channel counting or muliple-channel counting with quadrature.
Encoder output	The type of electrical signal that will be produced: Digital Square Wave, Analog Voltage, Analog Current, Serial (RS232, RS422, and so forth.), Parallel (GPIB), Serial Synchronous Interface (SSI), Ethernet, CANbus, and so forth.

Types of encoders

Position Encoders are used to obtain position, direction and speed information from machines in order to determine the mechanical position of an object. This mechanical position is an "absolute position". They may also be used to determine a change in position between the encoder and object as well. The change in position in relation to the object and encoder would be an incremental change. Position Encoders are widely used in the industrial arena for sensing the position of tooling and multi-axis positioning. There are many different types of encoders, but they basically fall into four main sensing techniques. Those being:

- Mechanical
- Magnetic
- Optical
- Electromagnetic

Within those categories, there are two differing encoder measurement types:

- Absolute
- Incremental (some incremental encoders are classified as quadrature encoders)

An encoder with an optical sensing type interprets data in pulses of light which can then be used to determine such things as position, direction and velocity. The shaft rotates a disc with opaque segments that represent a particular pattern. These encoders can determine the movement of an object for "rotary" or "shaft" applications while determining exact position in "linear" functions. This encoder sensing technique can be used in various applications such as printers, CNC milling machines and robotics.

Description of encoders

Linear encoders

The Linear Encoder uses a transducer to measure the linear distance between two points. These encoders can use a rod or a cable that is run between the encoder transducer and the object that will be measured for movement. As the object moves, the transducer's data collected from the rod or cable creates an output signal that is linear to the object's movement. As the distance is measured, the Linear Encoder uses this information to determine the position of the object.

An example of where a linear encoder may be used is for a CNC milling machine where precise movement measurements are required for accuracy in manufacturing. Linear Encoders can be "Absolute" or "Incremental".

Rotary encoders

A Rotary encoder collects data and provides feedback based on the rotation of an object or in other words, a rotating device. Rotary Encoders are sometimes called "Shaft Encoders". This encoder type can convert an object's angular position or motion based on the rotation of the shaft, depending on the measurement type used.

"Absolute Rotary Encoders" can measure "angular" positions while
"Incremental Rotary Encoders" can measure things such as distance, speed and position.

Rotary Encoders are employed in a wide variety of application areas such as computer input device like mice and trackballs as well as robotics.

Position encoders

A Position Encoder is used to determine the mechanical position of an object. This mechanical position is an "absolute position". They may also be used to determine a change in position between the encoder and object as well. The change in position in relation to the object and encoder would be an incremental change.
Position Encoders are widely used in the industrial arena for sensing the position of tooling and multi-axis positioning.

Optical encoders

An Optical Encoder interprets data in pulses of light which can then be used to determine such things as position, direction and velocity. The shaft rotates a disc with opaque segments that represent a particular pattern. These encoders can determine the movement of an object for "rotary" or "shaft" applications while determining exact position in "linear" functions. Optical encoders are used in various applications such as printers, CNC milling machines and robotics.

Absolute Vs incremental encoders

To demonstrate the difference between Absolute and Incremental Encoders we will use the Rotary Encoder type as an example.

Absolute rotary encoders

Absolute rotary encoders are capable of providing unique position values from the moment they are switched on. This is accomplished by scanning the position of a coded element. Each position in these systems correspond to a unique code. The number of codes, n, per revolution corresponds to the resolution or 2^n unique positions.

- Multiple Interface Options: Analog, Ethernet, Fieldbus, Parallel, Serial
- Singleturn and Multiturn Revolution
- Optical, Mechanical, Magnetic, and Electromagnetic Measuring Principles

Incremental encoders

Incremental encoders generate a pulse train output signal each time the shaft rotates. The number of pulses per revolution defines the resolution of the device, the number of pulses indicates the change in angle, the frequency of pulses is proportional to the rate of position change, and the phase indicates the direction of movement. Each time the encoder is powered on it begins counting from zero, regardless of where the shaft is. Initial homing to a reference point, often called an index or Z signal, is therefore inevitable in all positioning tasks, both upon start up of the control system and whenever power to the encoder has been interrupted.

- QEPA, QEPB, QEPI (Index), and inverted Signals are output of the Incremental Encoders
- Flexible Scaling Functionality

Figure 34 is an example of a quadrature signal moving in the forward direction (QEPA leads QEPB).

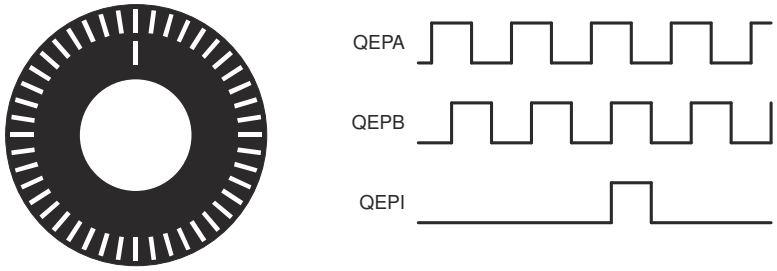
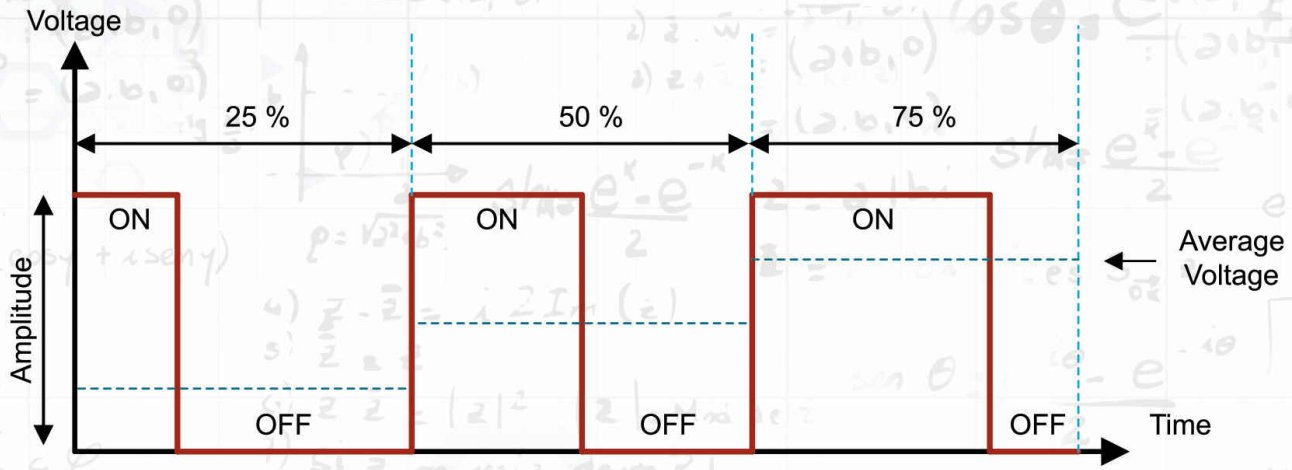


Figure 34. *Quadrature signal.*

Notes

Pulse width modulation (PWM)



PWM definitions

Period (T) = 1/F	The duration after which the PWM pattern will repeat itself
Frequency (F) = 1/T	Inverse of the period
On-Time	The amount of time the output signal has a logic high value during a period
Off-Time	The amount of time the output signal has a logic low value during a period
Amplitude	The max voltage of the output signal

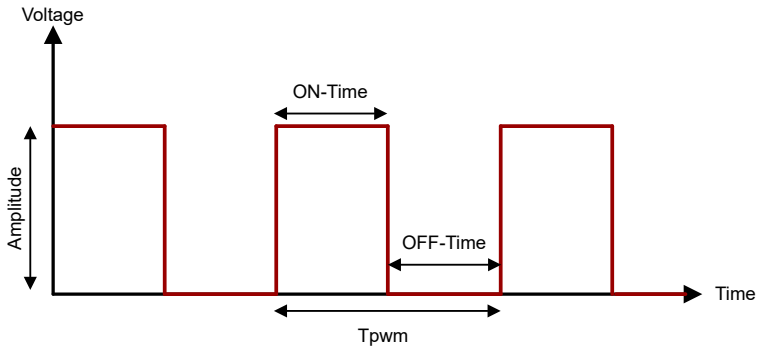


Figure 35. *PWM output.*

Duty cycle

Duty cycle describes the portion of the PWM period that is 'on' or high. The duty cycle is measured as a percentage.

$$\text{Duty Cycle } (\%) = \frac{\text{"ON Time"}}{\text{Period}} * 100 \tag{81}$$

$$\text{Average Output Voltage } (\text{V}) = \frac{\text{"ON Time"}}{\text{Period}} * \text{ Amplitude} \tag{82}$$

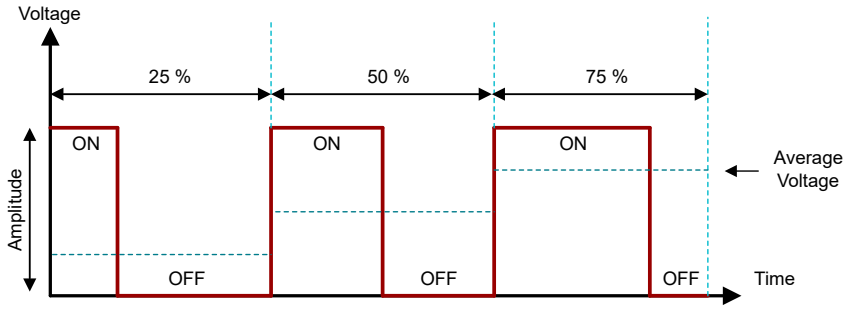


Figure 36. *PWM duty cycle.*

Example

What is the duty cycle for an average voltage output of 2.75 V if the amplitude is 3 V and the period is 200 ms?

Answer

$$\text{Average Output Voltage} = \frac{\text{"ON Time"}}{\text{Period}} * \text{Amplitude} = 2.75 = \frac{\text{"ON Time"}}{200\text{ms}} * 3 \rightarrow \text{ON Time} = 183\text{ns} \quad (83)$$

$$\text{Duty Cycle (\%)} = \frac{\text{"ON Time"}}{\text{Period}} * 100 = \ = \frac{183\text{ms}}{200\text{ms}} * 100 = 91.5\% \quad (84)$$

Resolution

$$\text{PWM resolution (\%)} = \frac{F_{PWM}}{F_{PWMCLK}} * 100\% \quad (85)$$

$$\text{PWM resolution (bits)} = \log_2\left(\frac{T_{PWM}}{T_{PWMCLK}}\right) \quad (86)$$

Where

$PWM_{resolution}$ = the granularity with which the duty cycle can be modulated

F_{PWM} = frequency of PWM output, $1/\ T_{PWM}$

F_{PWMCLK} = frequency of PWM clock, $1/\ T_{PWMCLK}$

T_{PWM} = period of PWM output, $1/\ F_{PWM}$

T_{PWMCLK} = period of PWM clock, $1/\ F_{PWMCLK}$

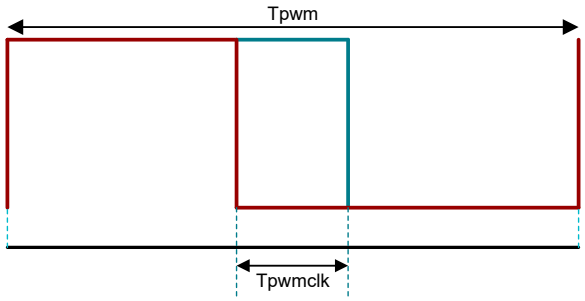


Figure 37. *PWM resolution.*

Table 17. *Common PWM resolution values.*

PWM Frequency (kHz)	Resolution (bits)	Resolution (%)
20	12.3	0.02
50	11	0.05
100	10	0.1
150	9.4	0.15
200	9	0.2
250	8.6	0.25
500	7.6	0.5
1000	6.6	1
1500	6.1	1.5
2000	5.6	2

Example

What is the resolution of a PWM output that has a 500 ns period and a PWM clock frequency of 200 MHz?

Answer

$$\text{PWM resolution (\%)} = \frac{F_{PWM}}{F_{PWMCLK}} * 100\% = \frac{\frac{1}{500ns}}{200MHz} * 100 = 1\ \%$$

$$\text{PWM resolution (bits)} = \log_2\left(\frac{T_{PWM}}{T_{PWMCLK}}\right) = \log_2\left(\frac{500ms}{\frac{1}{200MHz}}\right) = 6.64 \text{ bits} \rightarrow 6 \text{ bits}$$

Deadband

Figure 38 illustrates a PWM output with both rising and falling edge delays applied. This method provides a means to delay the switching of a gate signal, thereby allowing time for gates to turn off and preventing a short circuit. To explain further, power-switching devices turn on faster than they shut off. This issue would momentarily provide a path from supply rail to ground, resulting in a short circuit. The separation of PWM signal transitions between the outputs is referred to as "dead-band"

- Rising edge delay (RED) is a delay at the rising edge of the output
- Falling edge delay (FED) is a delay at the falling edge of the output

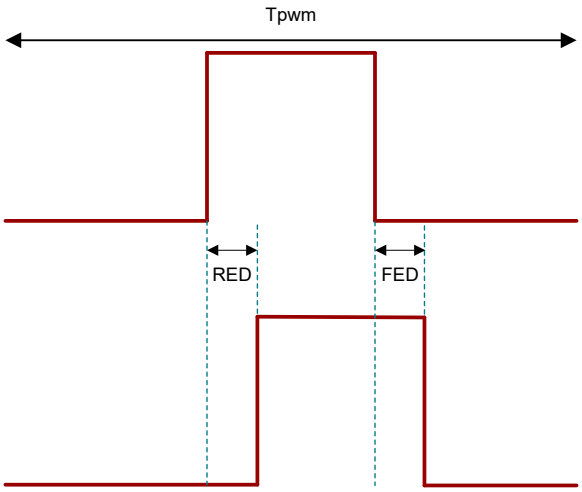
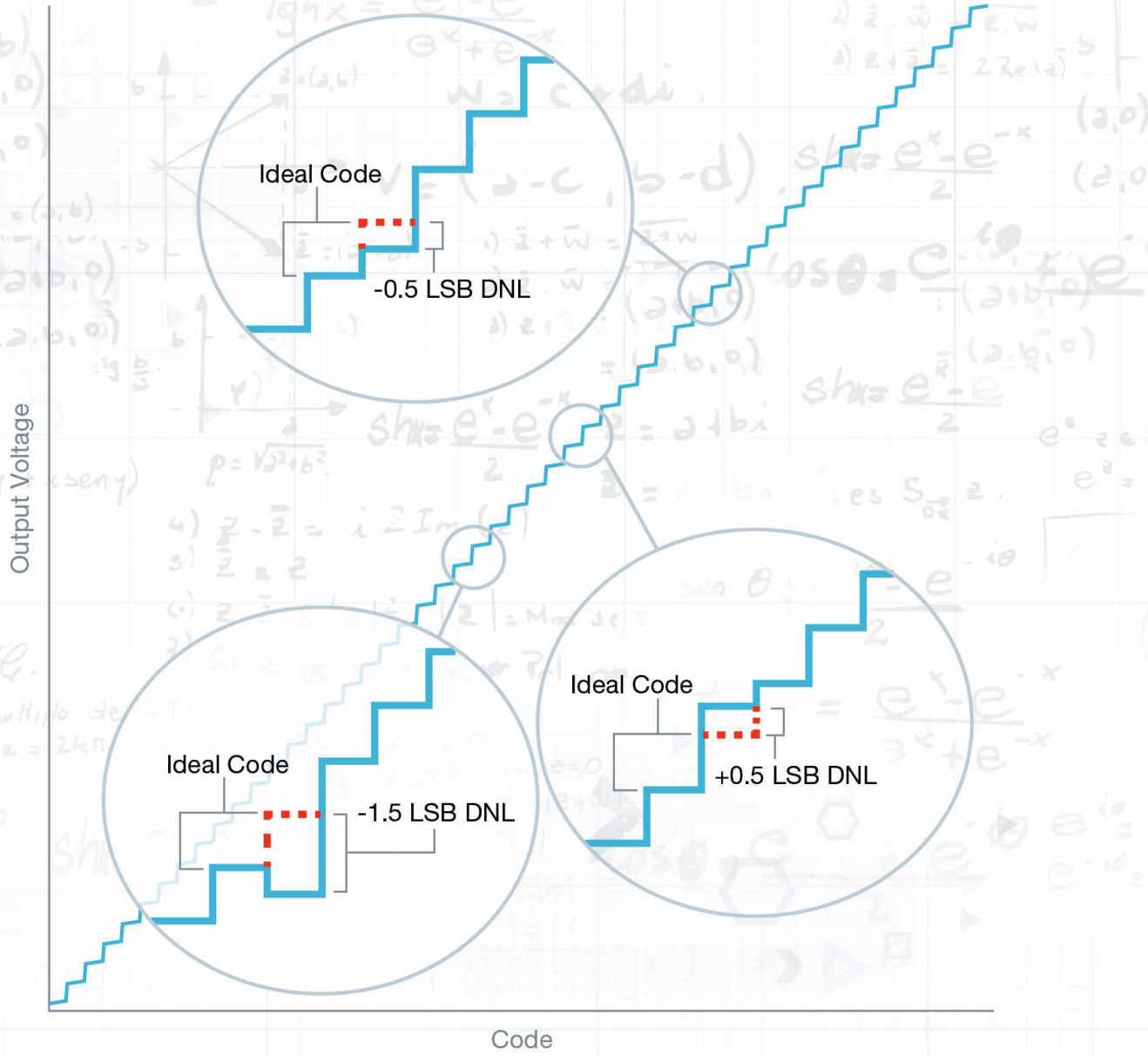


Figure 38. *PWM deadband.*

Notes

DAC



DAC definitions

Resolution = n	The number of bits used to represent each input code
Number of codes = 2^n	The number of possible codes that can be enumerated
Full-scale range output = FSR	The limit of the output voltage range
$\text{LSB} = \frac{\text{FSR}}{2^n}$	The voltage difference between two consecutive codes
Full-scale output voltage = $(2^n - 1) \bullet 1\text{LSB}$	The maximum voltage that can be output
Full-scale input code = $2^n - 1$	The largest code that can be represented
Transfer function: $V_{Out} = \frac{FSR}{2^n} \bullet Code$	Relationship between input code and output voltage

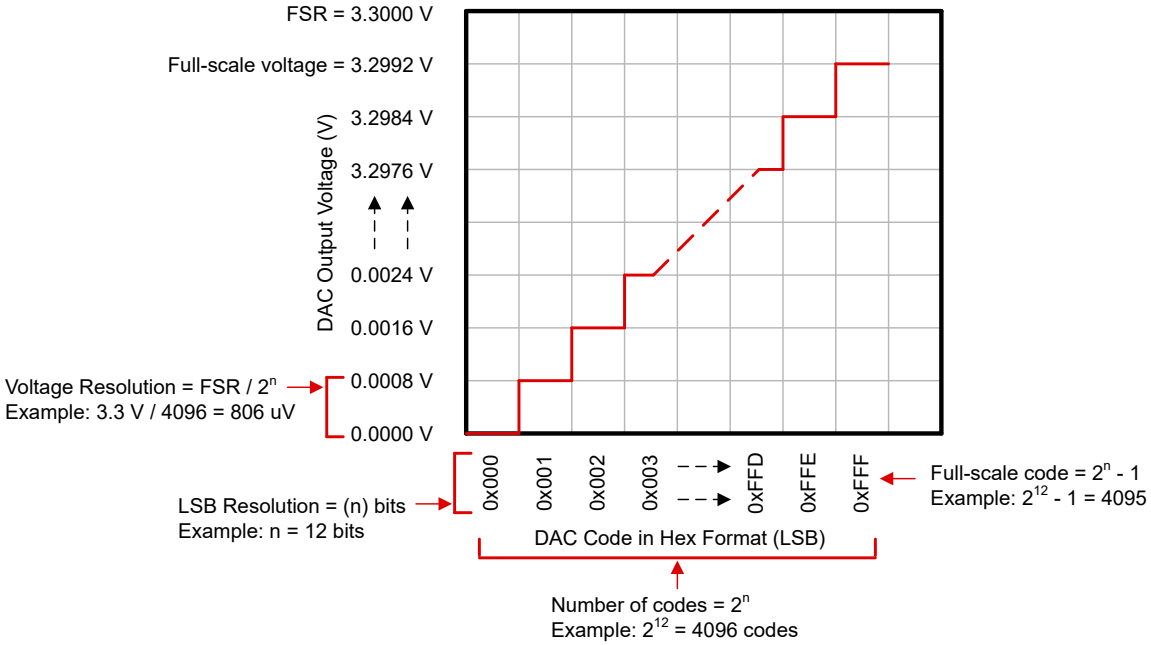


Figure 39. *DAC transfer function.*

DAC error

DAC offset error

Offset error describes the deviation from the ideal y-axis intercept for the DAC transfer function. The effect of offset error applies uniformly across the output range, and does not include the contribution of gain error or any nonlinear errors that may be present in zero-code error.

For an ideal transfer function described as:

$$V_{Out} = (\frac{FSR}{2^n} \bullet Code) \tag{87}$$

The effect of offset error (E_{Offset}) is modeled as:

$$V_{Out} = (\frac{FSR}{2^n} \bullet Code) + E_{Offset} \tag{88}$$

Offset error may be derived using different techniques such as a simple mid-point measurement, or a calculated y-axis intercept based on a best fit line that is extrapolated from multiple measured points.

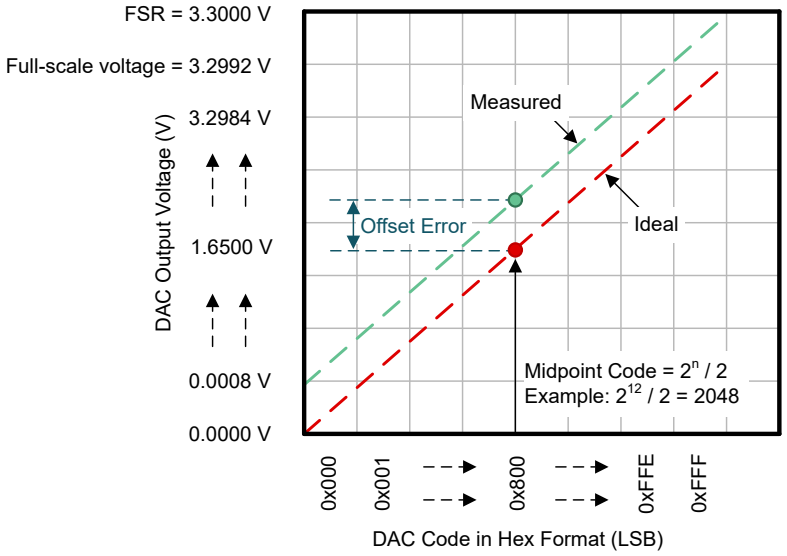


Figure 40. *DAC midpoint offset error.*

DAC gain error

Gain error describes the deviation from the ideal slope of the DAC transfer function. The effect of gain error scales in magnitude across the output range, and does not include the contribution of offset error or any nonlinear errors that may be present in full-scale error.

For an ideal transfer function described as:

$$V_{Out} = (\frac{FSR}{2^n} \bullet Code) \tag{89}$$

The effect of gain error (E_{Gain}) is modeled as:

$$V_{Out} = (\frac{FSR}{2^n} \bullet Code) \bullet E_{Gain} \tag{90}$$

Gain error is derived from the measured values of multiple points in the transfer function. The methodology used for extrapolating the gain error may vary between devices.

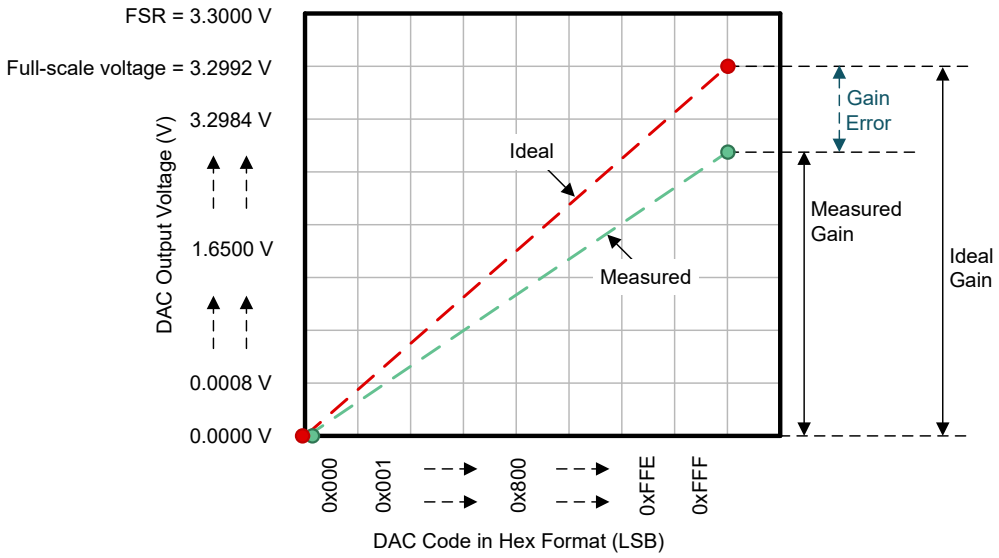


Figure 41. *DAC gain error.*

DAC zero-code error

Zero-Code error is an end-point error for when Code = 0 is loaded into the DAC code register. The intention is to describe how close to the zero voltage the DAC output can reach when set to the minimal output value. Zero-Code error is correlated to the cumulative effects of offset error and linearity error.

A positive error indicates that the measured output voltage for Code = 0 is higher than ideal, whereas a negative error indicates that the measured output voltage for Code = 0 is lower than ideal.

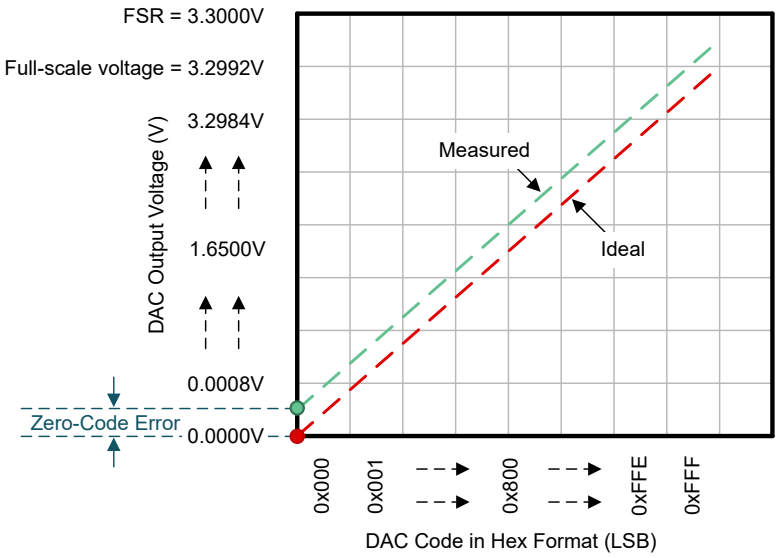


Figure 42. *DAC xero-code error.*

DAC full-scale error

Full-Scale error is an end-point error for when Code = 2^n-1 is loaded into the DAC code register. The intention is to describe how close to the full-scale voltage the DAC output can reach when set to the maximum output value. Full-Scale error is correlated to the cumulative effects of gain error and linearity error.

A negative error indicates that the measured output voltage for Code = 2^n-1 is lower than ideal, whereas a positive error indicates that the measured output voltage for 2^n-1 is higher than ideal.

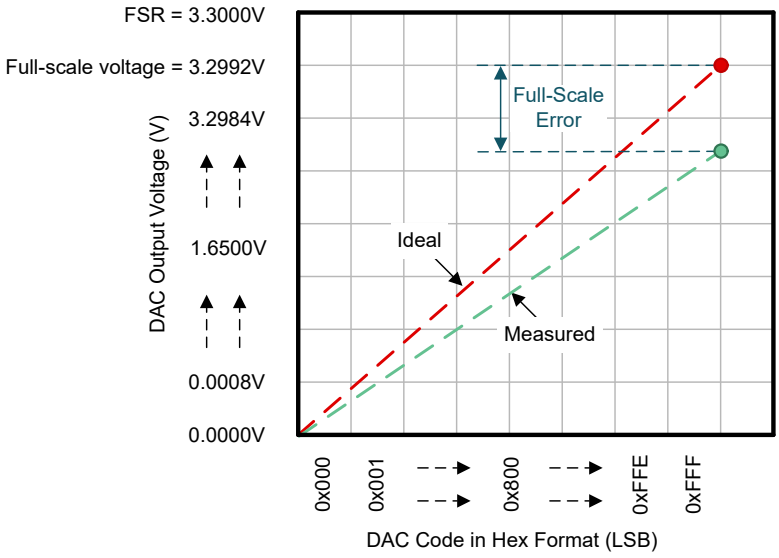


Figure 43. *DAC full-scale error.*

DAC differential non-linearity (DNL)

Differential Non-Linearity (DNL) describes the difference between measured versus ideal step sizes between sequential codes. The DAC manual will typically express only the minimum and maximum deviations for the linear region of the transfer function. DNL performance is typically at its worst near the power supply rails due to internal component saturation. Most modern DACs are monotonic, meaning that the output voltage will not decrease when the input code is incremented. **Figure 44** shows both monotonic and non-monotonic DNL.

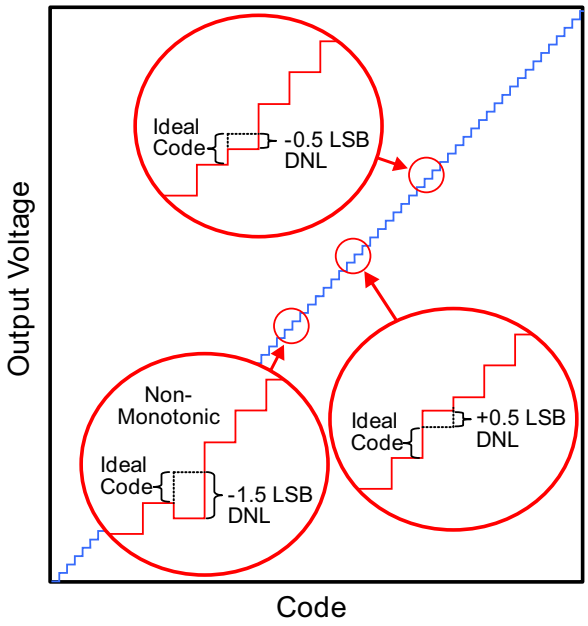


Figure 44. *DAC DNL.*

DAC integral non-linearity (INL)

Integral Non-Linearity (INL), sometimes referred to as relative accuracy, describes the deviation of the measured output versus a straight line fit of the transfer function. While DNL expresses the relationship between measured vs ideal code step-sizes, INL expresses the cumulative effects of sequential DNL errors. The DAC manual will typically express only the minimum and maximum deviations for the linear region of the transfer function. INL error cannot be corrected using a simple two point fit calibration.

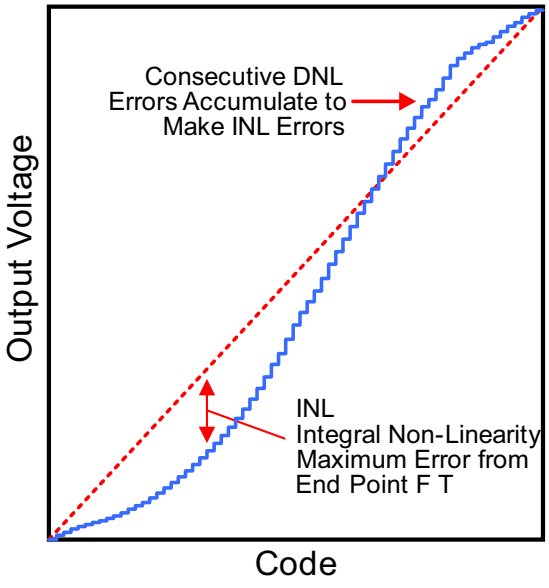


Figure 45. *DAC INL.*

DAC total unadjusted error (TUE)

The Total Unadjusted Error (TUE) is the statistical combination of uncorrelated error sources in the linear region of operation for the DAC. **Table 18** shows the correlative relationships between the various DAC errors that are defined in this chapter.

Table 18. *DAC error correlation.*

Error	Offset	Gain	Zero-Code	Full-Scale	DNL	INL
Offset	-	-	Correlated	Correlated	-	-
Gain	-	-	-	Correlated	-	-
Zero-Code	Correlated	-	-	-	-	-
Full-Scale	Correlated	Correlated	-	-	Correlated	Correlated
DNL	-	-	-	Correlated	-	Correlated
INL	-	-	-	Correlated	Correlated	-

The **TUE equation** is shown below, where all error sources must first be normalized to a common unit format (such as LSBs or parts per million). **Table 19** shows the calculations required to convert between different unit formats.

TUE Equation

$$TUE = \sqrt{E_{Offset}{}^2 + E_{Gain}{}^2 + E_{INL}{}^2} \quad (91)$$

Where

E_{Offset} = The static component of output error across the transfer function. See **DAC Offset Error**.

E_{Gain} = The proportionate component of output error across the transfer function. See **DAC Gain Error**.

E_{INL} = The maximum deviation of the output from a straight-line fit of the transfer function. See **DAC INL**.

Table 19. *Unit conversions for error.*

Convert		To			
		Codes	**Volts**	**%**	**ppm**
From	**Codes**	-	$\frac{Codes \bullet V_{FSR}}{2^n}$	$\frac{Codes \bullet 100}{2^n}$	$\frac{Codes \bullet 10^6}{2^n}$
	Volts	$\frac{Volts \bullet 2^n}{V_{FSR}}$	-	$\frac{Volts \bullet 100}{V_{FSR}}$	$\frac{Volts \bullet 10^6}{V_{FSR}}$
	%	$\frac{\% \bullet 2^n}{100}$	$\frac{\% \bullet V_{FSR}}{100}$	-	$\frac{\% \bullet 10^6}{100}$
	ppm	$\frac{ppm \bullet 2^n}{10^6}$	$\frac{ppm \bullet V_{FSR}}{10^6}$	$\frac{ppm \bullet 100}{10^6}$	-

A single TUE calculation can be useful for comparing the relative performance between different DACs, but it may not produce accurate estimates of typical error in the system. For example, the TUE equation treats E_{Gain} as a uniform contributor of error across the transfer function, but E_{Gain} is actually a scaled error with minimal influence on the lower codes. An improved estimate of system error may be produced by breaking up the transfer function into multiple regions, where the error components are adjusted in the TUE calculation based on their expected contribution to each region.

DAC output considerations

DAC linear range

Linear range defines the region of the DAC transfer function that is expected to follow a uniform slope, where the range can be described using either the code range or the output voltage range. The linear range for a DAC is often the same as its full programmable range.

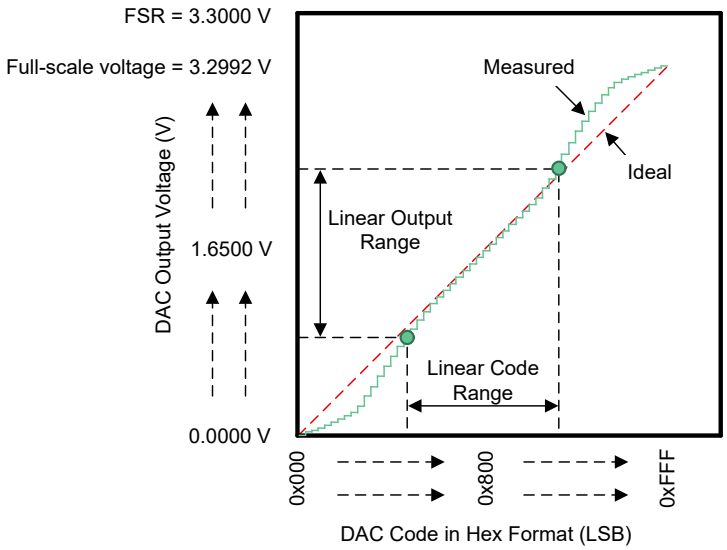


Figure 46. *DAC linear range.*

If the DAC linear range is a subset of the full programmable range, some error parameters (such as those relating to gain and linearity) may apply to the linear range only. These error parameters may include additional qualifiers such as end point correction. For example, **Table 20** shows how the straight line reference used for calculating the INL error may be derived from measured values as opposed to ideal values.

Table 20. *Straight line end points for INL.*

End Points	Ideal Line	End Point Corrected Line
Min Code	Code = 0	Minimum Linear Code
Min Voltage	Ideal Voltage	Measured Voltage
Max Code	Code = 2^n-1	Maximum Linear Code
Max Voltage	Ideal Voltage	Measured Voltage

DAC settling time

The settling time describes the speed at which the DAC output will reach a known and useful level after the input code is modified. If the DAC is used as a static reference level, the settling time may only be a minor consideration that applies once during system initialization. For systems that require a dynamic DAC output, the settling time may instead be a major consideration for meeting real-time deadlines.

Figure 47 shows a preconditioned DAC output of V0 before the input DAC code is changed at time T0. The new DAC code has a target output value of V2 that will require some amount of full settling time (T2 - T0) to reach. In the time between T0 and T2, the DAC output would typically be described as unstable.

If the full settling time (T2 - T0) is considered to be too slow for practical usage, an intermediate settling time (T1 - T0) may be provided with a bounded expectation for settling error (V1 - V2).

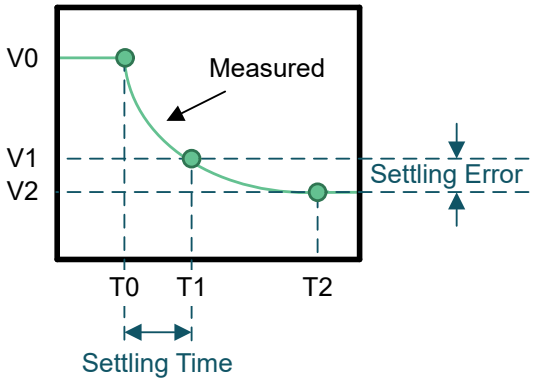


Figure 47. *DAC settling time.*

DAC load regulation

Load regulation describes the ability of the DAC output to drive an electrical load while still meeting its performance specifications. The maximum load is often expressed in terms of a minimum resistance and maximum capacitance of a low-pass RC filter configuration as shown in **Figure 48**.

A resistive load that is less than R_{MIN} may overwhelm the drive strength of the DAC output. A capacitive load that is greater than C_{MAX} may lead to instability of the DAC output.

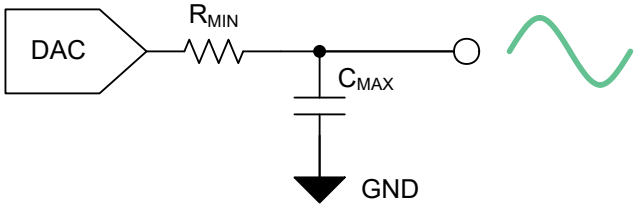
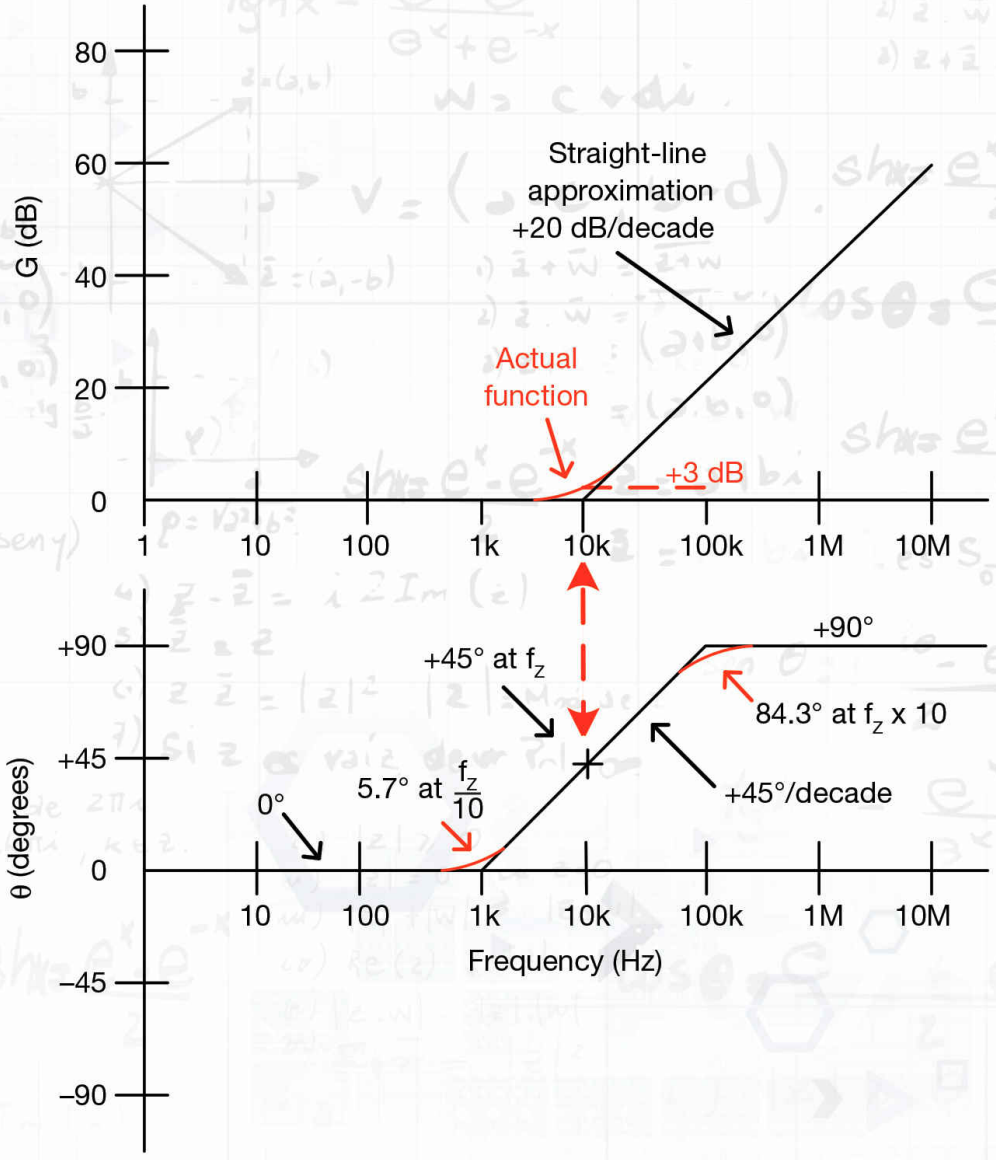


Figure 48. *DAC load regulation.*

Notes

Mathematical models



Laplace transforms

Linear system analysis is facilitated by the use of Laplace transforms.

If f(t) is a real function of time defined for all t > 0, its Laplace tansform f(s) is,

$$f(s) = L\{f(t)\} = \int_{0}^{\infty+} f(t)e^{-st}dt \tag{92}$$

Table 21. *Important Laplace transform pairs.*

f(t)	f(s)
Step function, $u(t)$	$\frac{1}{s}$
$e^{-\alpha t}$	$\frac{1}{s+\alpha}$
$\sin(\omega t)$	$\frac{\omega}{s^2+\omega^2}$
$\cos(\omega t)$	$\frac{s}{s^2+\omega^2}$
t^n	$\frac{n!}{s^{n+1}}$
$f^{(k)}(t) = \frac{d^k f(t)}{dt^k}$	$s^k F(s) - s^{k-1} f(0'') - s^{k-2} f'(0'') - \ldots - f^{(k-1)}(0'')$
$\int_{-\infty}^{t} f(t)dt$	$\frac{F(s)}{s} + \frac{1}{s}\int_{-\infty}^{0} f(t)dt$
Impulse function $\delta(t)$	1
$e^{-\alpha t}\sin(\omega t)$	$\frac{\omega}{(s+\alpha)^2+\omega^2}$
$e^{-\alpha t}\cos(\omega t)$	$\frac{s+a}{(s+\alpha)^2+\omega^2}$
$\frac{1}{\omega}\left[(\alpha-\alpha)^2+\omega^2\right]^{1/2} e^{-\alpha t}\sin(\omega t + \emptyset)$ $\emptyset = \tan^{-1}\left(\frac{\omega}{-\alpha}\right)$	$\frac{s+a}{(s+\alpha)^2+\omega^2}$
$\frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t}\sin\left(\omega_n\sqrt{1-\zeta^2}t\right),\ \zeta < 1$	$\frac{{\omega_n}^2}{s^2+2\zeta\omega_n s+{\omega_n}^2}$
$\frac{1}{\alpha^2+\omega^2} + \frac{1}{\omega\sqrt{\alpha^2+\omega^2}} e^{-at}\sin(\omega t - \emptyset)$ $\emptyset = \tan^{-1}\left(\frac{\omega}{-\alpha}\right)$	$\frac{1}{s\left[(s+\alpha)^2+\omega^2\right]}$

Transfer function

$$a_n y^{(n)}(t) + \ldots + a_1 y'(t) + a_0 y(t) = b_m u^{(m)}(t) + \ldots + b_1 u'(t) + b_0 u(t)$$

For initial zero conditions, the differential equation can be written in Laplace form as,

$$a_n s^n y(s) + \ldots + a_1 s y(s) + a_0 y(s) = b_m s^m u(s) + \ldots + b_1 s u(s) + b_0 u(s)$$

$$(a_n s^n + \ldots + a_1 s + a_0) y(s) = (b_m s + \ldots + b_1 s + b_0) u(s)$$

$$\alpha(s) y(s) = \beta(s) u(s)$$

The dynamic behavior of a system is characterized by the roots of $\beta(s)$ and $\alpha(s)$

The m roots of $\beta(s)$ are called the **zeros** of the system,

$$\beta(s) = b_m s^m + ... + b_1 s + b_0 \tag{93}$$

The n roots of $\alpha(s)$ are called the **poles** of the system,

$$\alpha(s) = a_n s^n + ... + a_1 s + a_0 \tag{94}$$

The ratio $\frac{\beta(s)}{\alpha(s)}$ is the **transfer function** of the system

$$G(s) = \frac{y(s)}{u(s)} = \frac{\beta(s)}{\alpha(s)} = \frac{b_m s^m + \ldots + b_1 s + b_0}{a_n s^n + \ldots + a_1 s + a_0} \tag{95}$$

The transfer function of a system is the Laplace transform of its impulse response,

$$y(t) = g(t)^* u(t) = L^{-1}\{G(s)u(s)\} \tag{96}$$

Transient response

The numerator and denominator of a transfer function, $G(s)$ can be factorized to express the transfer function itself in terms of poles and zeros,

$$y(s) = A\frac{(s - z_1)(s - z_2)\ldots(s - z_m)}{(s - p_1)(s - p_2)\ldots(s - p_n)}u(s) \tag{97}$$

This rational function yields q terms, state vectors, through partial fraction expansion,

$$y(s) = \frac{\varepsilon_1}{s - r_1} + \frac{\varepsilon_2}{s - r_2} + \ldots + \frac{\varepsilon_q}{s - r_q} \tag{98}$$

The time response is a sum of exponential terms, where each index is a denominator root,

$$y(t) = \varepsilon_1 e^{r1t} + \varepsilon_2 e^{r2t} + \ldots + \varepsilon_n e^{rnt} + \varepsilon_{n+1} e^{rn+1t} + \ldots + \varepsilon_q e^{rqt} \tag{99}$$

Table 22. *Response portions.*

Response Type	Representation	Value	Description
Transient Response	$y_c(t)$	$\varepsilon_1 e^{r1t} + \varepsilon_2 e^{r2t} + \ldots + \varepsilon_n e^{rnt}$	The n terms in y(t) with roots originating from comprise the transient response
Steady State Response	$y_p(t)$	$\varepsilon_{n+1} e^{rn+1t} + \ldots + \varepsilon_q e^{rqt}$	The a-n terms originating from u(s) comprise the steady state response

Frequency response

If a steady state sine wave, $u(t) = u_0 \sin(\omega t + \alpha)$ is applied to a linear system denoted by $G(s)$, the linear system would respond at the same frequency with a certain phase and magnitude, giving output $y(t) = y_0 \sin(\omega t + \beta)$. The amplitude is modified by $\frac{y_0}{u_0}$ and the phase is shifted by $\emptyset = \beta - \alpha$ or $< G(j\omega)$

Bode plot basics

The frequency response for the magnitude or gain plot is the change in voltage gain as frequency changes. The change is specified on a Bode plot, a plot of frequency versus voltage gain in dB (decibles). Bode plots are usually plotted as semi-log plots with frequency on the x-axis, log scale, and gain on the y-axis, linear scale. The other half of the frequency response is the phase shift versus frequency and is plotted as frequency versus degree phase shift. Phase plots are usually plotted as semi-log plots with frequency on the x-axis, log scale, and phase shift on the y-axis, linear scale.

Definitions

Voltage gain in decibels

$$\text{Voltage gain (dB)} = 20 \log\left(\frac{V_{OUT}}{V_{IN}}\right) \tag{100}$$

Power gain in decibels

$$\text{Power gain (dB)} = 10 \log\left(\frac{P_{OUT}}{P_{IN}}\right) \tag{101}$$

Used for input or output power

$$\text{Power measured (dBm)} = 10 \log\left(\frac{\text{Power measured (W)}}{1\ \text{mW}}\right) \tag{102}$$

Table 23. *Examples of common gain values and dB equivalent.*

A (V/V)	A (dB)
0.001	-60
0.01	-40
0.1	-20
1	0
10	20
100	40
1,000	60
10,000	80
100,000	100
1,000,000	120
10,000,000	140

Where

Roll-off rate is the decrease in gain with frequency

Decade is a tenfold increase or decrease in frequency (from 10 Hz to 100 Hz is one decade)

Octave is the doubling or halving of frequency (from 10 Hz to 20 Hz is one octave)

Bode plots: Poles

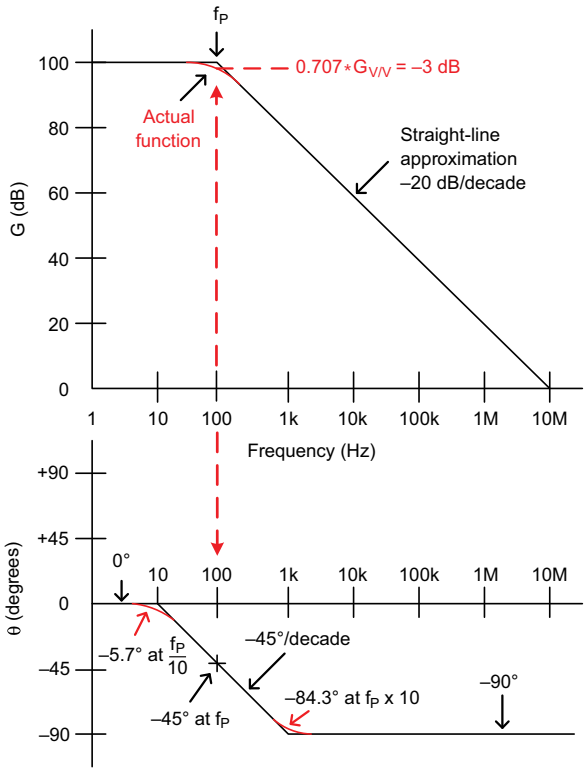


Figure 49. *Pole gain and phase.*

Where

Pole location = f_p (cutoff frequency)

Magnitude $\left(f < f_p\right)$ = G_{DC} (for example, 100 dB)

Magnitude $\left(f = f_p\right)$ = -3 dB

Magnitude $\left(f > f_p\right)$ = -20 dB/decade

Phase $\left(f = f_p\right) = -45°$

Phase $\left(0.1\, f_p < f < 10\, f_p\right) = -45°$/decade

Phase $\left(f > 10\, f_p\right) = -90°$

Phase $\left(f < 0.1\, f_p\right) = 0°$

Pole (equations)

As a complex number

$$G_V = \frac{V_{OUT}}{V_{IN}} = \frac{G_{DC}}{j\left(\frac{f}{f_p}\right) + 1} \tag{103}$$

Magnitude

$$G_V = \frac{V_{OUT}}{V_{IN}} = \frac{G_{DC}}{\sqrt{\left(\frac{f}{f_p}\right)^2 + 1}} \quad (104)$$

Phase Shift

$$\theta = -\tan^{-1}\left(\frac{f}{f_p}\right) \quad (105)$$

Magnitude in dB

$$G_{dB} = 20\text{Log}(G_V) \quad (106)$$

Where

G_V = voltage gain in V/V

G_{DB} = voltage gain in decibels

G_{DC} = the dc or low frequency voltage gain

f = frequency in Hz

f_p = frequency at which the pole occurs

θ = phase shift of the signal from input to output

j = indicates imaginary number or $\sqrt{-1}$

Bode plot (zeros)

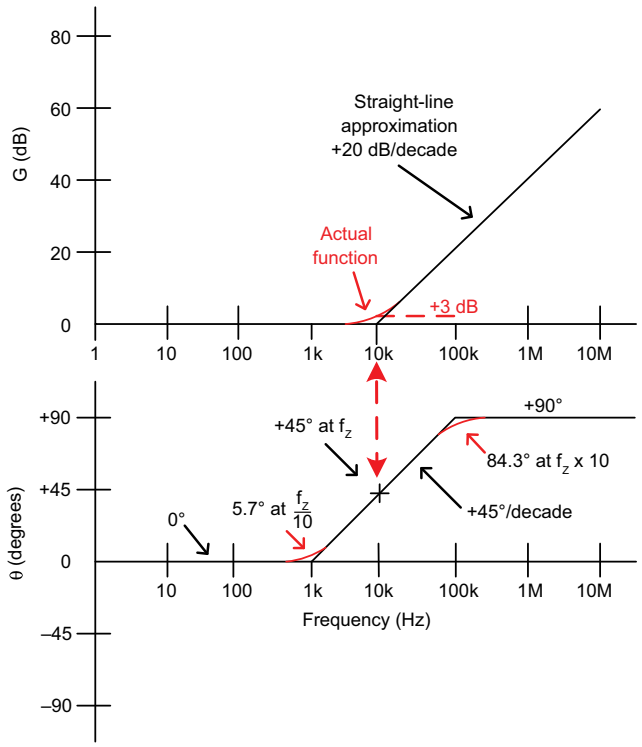


Figure 50. *Zero gain and phase.*

Where

Zero location = f_z

Magnitude $(f < f_z)$ = 0 dB

Magnitude $(f = f_z)$ = +3 dB

Magnitude $(f > f_z)$ = +20 dB/decade

Phase $(f = f_z) = +45°$

Phase $(0.1\, f_z < f < 10\, f_z) = +45°$/decade

Phase $(f > 10\, f_z) = +90°$

Phase $(f < 0.1\, f_z) = 0°$

Zero (equations)

As a complex number

$$G_V = \frac{V_{OUT}}{V_{IN}} = G_{DC}\left[j\left(\frac{f}{f_p}\right) + 1\right] \quad (107)$$

Magnitude

$$G_V = \frac{V_{OUT}}{V_{IN}} = G_{DC}\sqrt{\left(\frac{f}{f_z}\right)^2 + 1} \quad (108)$$

Phase Shift

$$\theta = \tan^{-1}\left(\frac{f}{z}\right) \quad (109)$$

Magnitude in dB

$$G_{dB} = 20\mathrm{Log}(G_V) \quad (110)$$

Where

G_V = voltage gain in V/V

G_{DB} = voltage gain in decibels

G_{DC} = the dc or low frequency voltage gain

f = frequency in Hz

f_z = frequency at which the zero occurs

θ = phase shift of the signal from input to output

j = indicates imaginary number or $\sqrt{-1}$

Z-domain

Definition of Bilateral Z-Transform

$$x(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \quad (111)$$

Definition of Unilateral Z-Transform

$$x(z) = \sum_{n=0}^{\infty} x[n]z^{-n} \quad (112)$$

Where:

x[n] = discrete time signal

x(z) = the z-domain transform of the discrete time signal

n = integer

z = complex number ($Ae^{j\Phi} = A(\cos\phi + j\sin\phi)$)

A = magnitude of z

j = imaginary unit

ϕ = phase in radians

Table 24. *Z-Transform Properties*

Property	Z-Transform
Linearity	$af_1[n] + bf_2[n] = aF_1(z) + bF_2(z)$
Shift left by k	$f[n+k] = z^k\left(F(z) - \sum_{n=0}^{k-1} f[n]z^{-n}\right)$
Shift right by k	$f[n-k] = z^{-k}F(z)$
Convolution	$f_1[n]*f_2[n] = F_1(z)F_2(z)$
Final Value Theorem	$\lim_{n \to \infty} f[n] = \lim_{z \to 1} (z-1)F(z)$

Notes

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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 will 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 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.

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

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265