**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 Circuits - 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

Message from the editors................................................................. 2
System Design................................................................................. 6
  Control......................................................................................... 7
  Feedback control....................................................................... 8
  Dynamic systems....................................................................... 9
  System stability......................................................................... 11
  Timing requirements................................................................... 12
  Discrete Time Domain................................................................. 15
  Filters......................................................................................... 17
Controllers..................................................................................... 23
  Linear PID.................................................................................. 24
  Linear PI.................................................................................... 26
  Nonlinear PID............................................................................ 27
  2P2Z.......................................................................................... 29
  3P3Z.......................................................................................... 30
  Direct form controllers............................................................... 31
ADC............................................................................................... 36
  ADC definitions......................................................................... 37
Table of Contents

ADC resolution.................................................................................................................. 37
Quantization error of ADC................................................................................................. 39
Total harmonic distortion (THD)......................................................................................... 40
AC signals........................................................................................................................... 42
DC signals.......................................................................................................................... 43
Settling time and conversion accuracy............................................................................... 44
ADC system noise.............................................................................................................. 45

Comparator......................................................................................................................... 47
Basic operation.................................................................................................................... 48
Offset and hysteresis.......................................................................................................... 48
Propagation delay............................................................................................................. 49

Processing.......................................................................................................................... 51
Data representation............................................................................................................ 52
Central processing unit....................................................................................................... 55
Memory............................................................................................................................... 56
Direct memory access (DMA)............................................................................................. 57
Interrupts............................................................................................................................. 58
Co-processors and accelerators......................................................................................... 58

Encoders............................................................................................................................. 61
Encoder definitions............................................................................................................ 62
Types of encoders.............................................................................................................. 62
Description of encoders..................................................................................................... 62
Absolute Vs incremental encoders..................................................................................... 63

Pulse width modulation (PWM).......................................................................................... 66
PWM definitions............................................................................................................... 67
Duty cycle......................................................................................................................... 67
Resolution......................................................................................................................... 68
Deadband.......................................................................................................................... 69

DAC.................................................................................................................................... 72
DAC definitions................................................................................................................. 73
DAC error.......................................................................................................................... 73
DAC output considerations............................................................................................... 78

Mathematical models........................................................................................................ 82
Laplace transforms........................................................................................................... 83
Transfer function.............................................................................................................. 83
Transient response............................................................................................................ 84
Frequency response......................................................................................................... 84
Z-domain.......................................................................................................................... 89

List of Figures

Figure 2-1. Control system................................................................................................. 7
Figure 2-2. Closed loop system........................................................................................ 7
Figure 2-3. Unit step response of a 1st order system (r=1)................................................... 10
Figure 2-4. Unit step response of an under-damped 2nd order system............................ 11
Figure 2-5. Digital control system.................................................................................... 16
Figure 2-6. Sampling......................................................................................................... 16
Figure 2-7. Filter response curves..................................................................................... 19
Figure 3-1. Parallel form linear PID controller................................................................. 24
Figure 3-2. Series form of linear PID controller............................................................... 25
Figure 3-3. PID control action......................................................................................... 26
Figure 3-4. Nonlinear PID input architecture.................................................................... 27
Figure 3-5. Nonlinear control law input-output plot......................................................... 27
Figure 3-6. Nonlinear PID linearized region...................................................................... 29
Figure 3-7. Phase and gain characteristics of a 2P2Z compensator................................... 29
Figure 3-8. Phase and gain characteristics of a 3P3Z compensator................................... 30
Figure 3-9. Representation of DF11.................................................................................. 31
Figure 3-10. Representation of DF13............................................................................... 32
Figure 3-11. Representation of DF22............................................................................... 33
Figure 3-12. Representation of DF23............................................................................... 34
Figure 4-1. ADC transfer function.................................................................................... 37
Figure 4-2. ADC full-scale range (FSR) unipolar............................................................. 37
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1-1</td>
<td>Characteristics of a good control system</td>
<td>7</td>
</tr>
<tr>
<td>Table 1-2</td>
<td>Rise and peak time for systems</td>
<td>12</td>
</tr>
<tr>
<td>Table 1-3</td>
<td>Damping ratios based on classification</td>
<td>14</td>
</tr>
<tr>
<td>Table 1-4</td>
<td>Filter types</td>
<td>17</td>
</tr>
<tr>
<td>Table 1-5</td>
<td>First order filters</td>
<td>19</td>
</tr>
<tr>
<td>Table 1-6</td>
<td>Second order filters</td>
<td>20</td>
</tr>
<tr>
<td>Table 1-7</td>
<td>Tuning parameter effect on the gain shape</td>
<td>28</td>
</tr>
<tr>
<td>Table 1-8</td>
<td>LSB voltage vs. resolution and reference voltage</td>
<td>39</td>
</tr>
<tr>
<td>Table 1-9</td>
<td>Conversion accuracy achieved after a specified time</td>
<td>44</td>
</tr>
<tr>
<td>Table 1-10</td>
<td>Time required to settle to a specified conversion accuracy</td>
<td>44</td>
</tr>
<tr>
<td>Table 1-11</td>
<td>Comparator inputs to output</td>
<td>48</td>
</tr>
<tr>
<td>Table 1-12</td>
<td>Primitive data types</td>
<td>52</td>
</tr>
<tr>
<td>Table 1-13</td>
<td>Multiplier prefixes and abbreviations</td>
<td>52</td>
</tr>
<tr>
<td>Table 1-14</td>
<td>ASCII table</td>
<td>52</td>
</tr>
<tr>
<td>Table 1-15</td>
<td>Real-time control processor - characteristics</td>
<td>55</td>
</tr>
<tr>
<td>Table 1-16</td>
<td>Memory types</td>
<td>57</td>
</tr>
<tr>
<td>Table 1-17</td>
<td>Common PWM resolution values</td>
<td>69</td>
</tr>
<tr>
<td>Table 1-18</td>
<td>DAC error correlation</td>
<td>78</td>
</tr>
<tr>
<td>Table 1-19</td>
<td>Unit conversions for error</td>
<td>78</td>
</tr>
<tr>
<td>Table 1-20</td>
<td>Straight line end points for INL</td>
<td>79</td>
</tr>
<tr>
<td>Table 1-21</td>
<td>Important Laplace transform pairs</td>
<td>83</td>
</tr>
<tr>
<td>Table 1-22</td>
<td>Response portions</td>
<td>84</td>
</tr>
<tr>
<td>Table 1-23</td>
<td>Examples of common gain values and dB equivalent</td>
<td>85</td>
</tr>
<tr>
<td>Table 1-24</td>
<td>Z-Transform Properties</td>
<td>90</td>
</tr>
</tbody>
</table>
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.

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

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition/Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stability</td>
<td>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.</td>
</tr>
<tr>
<td>Steady State Accuracy</td>
<td>How close the output is to the desired input. The closer the output is to the input the better the <strong>steady state accuracy</strong> is.</td>
</tr>
<tr>
<td>Satisfactory Transient Response</td>
<td>A <strong>transient response</strong> shows how the system behaves when it is taken out of equilibrium. Ideally, the system should not deviate from the steady state.</td>
</tr>
<tr>
<td>Satisfactory Frequency Response</td>
<td>A <strong>frequency response</strong> displays the output of a system in terms of magnitude and phase as a function of frequency.</td>
</tr>
<tr>
<td>Reduced Sensitivity to Disturbances</td>
<td>Disturbance in control systems signify any unwanted or unexpected inputs that alter the system’s output which cause an increase in the <strong>system error</strong>.</td>
</tr>
</tbody>
</table>

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 1. Control system.

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

$$e = r - Hy$$  \hspace{1cm} (1)

Output Equation

$$y = FGe$$  \hspace{1cm} (2)

Error and output equations combined

$$y (1 + FGH) = FG r$$  \hspace{1cm} (3)

Open loop transfer function

$$L = FGH$$  \hspace{1cm} (4)

Closed loop transfer function

$$\frac{y}{r} = \frac{FG}{1 + FGH}$$  \hspace{1cm} (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.

\[
\frac{e}{r} = \frac{1}{1 + FG} = \frac{1}{1 + L} = S
\]  

(6)

**Dynamic systems**

**First order system**

**Equations**

1st order differential equation

\[
\tau \frac{dy}{dt} + y(t) = u(t)
\]  

(7)

**Transfer function** of 1st order system

\[
\frac{y(s)}{u(s)} = \frac{1}{s\tau + 1}
\]  

(8)

Response of system

\[
y(t) = L^{-1}\left\{\frac{1}{s\tau + 1}u(s)\right\}
\]  

(9)

Response of system following a unit step response

\[
y(t) = 1 - e^{-\frac{t}{\tau}}
\]  

(10)

**Where**

\(\tau = \) time constant of the system

\(y(t) = \) output function

\(u(t) = \) input function
Figure 3. Unit step response of a 1st order system ($\tau=1$).

Second order system

Equations

2nd order differential equation

$$y''(t) + 2\zeta\omega_n y'(t) + \omega_n^2 y(t) = \omega_n^2 u(t)$$  \hspace{1cm} (11)

Transfer function of 2nd order system

$$\frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$  \hspace{1cm} (12)

Characteristic equation

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0$$  \hspace{1cm} (13)

Poles of a linear 2nd order system

$$s = -\zeta \omega_n \pm \omega_n \sqrt{\zeta^2 - 1}$$  \hspace{1cm} (14)
Unit step response of an under-damped 2nd order system

\[ y(t) = 1 - \frac{\omega_n}{\omega_d} e^{-\sigma t} \sin(\omega_d t + \varphi) \]  
\[ \varphi = \cos^{-1} \zeta \]

Where

\( y(t) = \) output function
\( u(t) = \) input function
\( \omega_n = \) un-damped natural frequency
\( \omega_d = \) damped natural frequency
\( \zeta = \) damping ratio

![Figure 4. Unit step response of an under-damped 2nd order system.](image)

**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 $\angle GH(j\omega) = -180^\circ$ 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:

$$\zeta = \frac{\text{PM}}{100}$$  \hspace{1cm} (17)

**Where**

$\zeta$ = damping ratio

PM = phase margin

**Timing requirements**

**Peak/rise time**

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

<table>
<thead>
<tr>
<th>System Order</th>
<th>System Classification</th>
<th>Rise Time %</th>
<th>Peak Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Order</td>
<td>-</td>
<td>10 - 90</td>
<td>Usually categorized as 2.2 $\tau$, where $\tau$ is the time constant</td>
</tr>
<tr>
<td>2nd Order</td>
<td>Under-damped</td>
<td>0 - 100</td>
<td>Time at which the step response reaches its maximum</td>
</tr>
<tr>
<td>2nd Order</td>
<td>Over-damped</td>
<td>10 - 90</td>
<td>Peak time is not defined</td>
</tr>
</tbody>
</table>

**Equation**

**Peak Time**

$$t_p = \frac{\pi}{\omega_n \zeta} = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}}$$  \hspace{1cm} (18)

**Where**

$\zeta$ = damping ratio

$\omega_n$ = 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

\[ 4\tau \quad (19) \]

Settling time of 2nd order system

\[ t_s = \frac{1}{\sigma} \ln \left( \frac{\zeta}{\sqrt{1 - \zeta^2}} \right) \quad (20) \]

\[ c = \frac{\omega_n}{\omega_d} \quad (21) \]

**Damping** coefficient

\[ \sigma = \zeta \omega_n \quad (22) \]

Logarithmic decrement

\[ \delta = \ln \left( \frac{x_0}{x_1} \right) \quad (23) \]

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

\[ \frac{c}{\delta} = \sqrt{1 - \zeta^2} \quad (24) \]

Where

\[ \tau = \text{time constant} \quad (\text{time it takes for the step response to reach } 63\% \text{ of its final value}) \]

\[ x_0/x_1 = \text{the amplitudes of two successive peaks within a step response} \]

\[ \omega_n = \text{un-damped natural frequency} \]

\[ \omega_d = \text{damped natural frequency} \]

\[ \zeta = \text{damping ratio} \]

**Overshoot**

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

Definitions

Peak response

\[ M_p = 1 + \exp \left( -\pi \zeta \sqrt{1 - \zeta^2} \right) \quad (25) \]

Percentage of overshoot

\[ \text{P.O.} = 100e^{-\pi \zeta / \sqrt{1 - \zeta^2}} \quad (26) \]
Damping ratio

\[ \zeta = \frac{-\ln \left( \frac{P_{0}}{P_{100}} \right)}{\sqrt{\pi^2 + \ln^2 \left( \frac{P_{0}}{P_{100}} \right)}} \]  

**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.**

<table>
<thead>
<tr>
<th>Damping Ratio</th>
<th>Roots</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \zeta &gt; 1 )</td>
<td>[ s = -\zeta \omega_n \pm \omega_n \sqrt{\zeta^2 - 1} ]</td>
<td>Over-damped</td>
</tr>
<tr>
<td>( \zeta = 1 )</td>
<td>[ s = -\omega_n ]</td>
<td>Critically damped</td>
</tr>
<tr>
<td>( 0 &lt; \zeta &lt; 1 )</td>
<td>[ s = -\zeta \omega_n \pm j\omega_n \sqrt{1 - \zeta^2} ]</td>
<td>Under-damped</td>
</tr>
<tr>
<td>( \zeta = 0 )</td>
<td>[ s = \pm j\omega_n ]</td>
<td>Un-damped</td>
</tr>
</tbody>
</table>

**Where**

\( \zeta = \) damping ratio

\( \omega_n = \) 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

\[ \mathcal{L}\{f(t - T)\} \]  

Delay in Laplace domain

\[ e^{-sT}f(s) \]  

Magnitude of delay

\[ |e^{-i\omega T}| = \sqrt{(\cos \omega T)^2 + (-\sin \omega T)^2} = 1 \]  

Phase of delay

\[ e^{-i\omega T} = \tan^{-1}\left(\frac{-\sin \omega T}{\cos \omega T}\right) = -\omega T \]

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.

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

Where

$e^{-sT} = \text{delay transfer function}$

**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<sub>T</sub>): 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** ($f_{-3dB}$): 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 $f_1$ and $f_2$ 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 referred to as the stopband. However, for pass-band or stop-band, and notch filters two stopbands exist.

• **Bandwidth** ($\beta$): 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, $Q = \frac{f_0}{f_2 - f_1}$

**Filter Types**

Table 4. Filter types.

<table>
<thead>
<tr>
<th>Type</th>
<th>Definition</th>
<th>Response Curve</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-Pass (HPF)</td>
<td>Removes low frequencies and retains high frequencies</td>
<td><img src="image" alt="Response Curve" /></td>
</tr>
<tr>
<td>Type</td>
<td>Definition</td>
<td>Response Curve</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Low-Pass (LPF)</td>
<td>Removes high frequencies and retains low frequencies</td>
<td><img src="chart1.png" alt="Low-Pass Filter" /></td>
</tr>
<tr>
<td>Band-Stop (BSF)</td>
<td>Removes an interval of frequencies with a band, retains others</td>
<td><img src="chart2.png" alt="Band-Stop Filter" /></td>
</tr>
<tr>
<td>Band-Pass (BPF)</td>
<td>Retains an interval of frequencies with a band, removes others</td>
<td><img src="chart3.png" alt="Band-Pass Filter" /></td>
</tr>
</tbody>
</table>
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.](image)

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.

<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Active/Passive</th>
<th>Circuit</th>
<th>Gain (Vout/Vin)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-Pass</td>
<td>Passive</td>
<td><img src="image" alt="Low-Pass Passive Circuit" /></td>
<td>( \frac{XC}{\sqrt{R^2 + XC^2}} )</td>
</tr>
<tr>
<td>High-Pass</td>
<td>Passive</td>
<td><img src="image" alt="High-Pass Passive Circuit" /></td>
<td>( \frac{R}{\sqrt{R^2 + XC^2}} )</td>
</tr>
<tr>
<td>Low-Pass</td>
<td>Active</td>
<td><img src="image" alt="Low-Pass Active Circuit" /></td>
<td>( \frac{XC}{\sqrt{R^2 + XC^2}} = 1 )</td>
</tr>
</tbody>
</table>
Filter Type | Active/Passive | Circuit | Gain (Vout/Vin)
--- | --- | --- | ---
High-Pass | Active | ![Circuit Diagram](image) | $\frac{R}{V} \frac{1}{R^2 + XRC} = 1$

**Where**

Transfer function of 1st order low pass filter

$$H(s) = \frac{\omega C}{s + \omega C}$$  \hspace{1cm} (32)

Transfer function of 1st order high pass filter

$$H(s) = \frac{s}{s + \omega C}$$  \hspace{1cm} (33)

Cutoff frequency

$$f_C = \frac{1}{2\pi RC}$$  \hspace{1cm} (34)
$$\omega_C = 2\pi f_C$$  \hspace{1cm} (35)

Capacitive reactance

$$X_C = \frac{1}{2\pi fC}$$  \hspace{1cm} (36)

**Table 6. Second order filters.**

| Filter Type | Active/Passive | Circuit | Gain (Vout/Vin)
--- | --- | --- | ---
Low-Pass | Passive | ![Circuit Diagram](image) | $\left(\frac{1}{\sqrt{2}}\right)^n$ where n is the number of stages
High-Pass | Passive | ![Circuit Diagram](image) |  |
<table>
<thead>
<tr>
<th>Filter Type</th>
<th>Active/Passive</th>
<th>Circuit</th>
<th>Gain (Vout/Vin)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low-Pass</td>
<td>Active</td>
<td><img src="image" alt="Low-Pass Circuit" /></td>
<td>$1 + \frac{R_A}{R_B}$</td>
</tr>
<tr>
<td>High-Pass</td>
<td>Active</td>
<td><img src="image" alt="High-Pass Circuit" /></td>
<td></td>
</tr>
</tbody>
</table>

### Where

Transfer function of 2nd order low pass filter

$$H(s) = \frac{\omega_C^2}{s^2 + 2\omega_C s + \omega_C^2}$$  \hspace{1cm} (37)

Transfer function of 2nd order high pass filter

$$H(s) = \frac{s^2}{s^2 + 2\omega_C s + \omega_C^2}$$  \hspace{1cm} (38)

Cutoff frequency (same capacitor and resistor values)

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

$$\omega_C = 2\pi f_C$$  \hspace{1cm} (40)

Cutoff frequency (different capacitor and resistor values)

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

$$\omega_C = 2\pi f_C$$  \hspace{1cm} (42)
Controllers

- Linear PID
- Linear PI
- Nonlinear PID
- 2P2Z
- 3P3Z
- Direct form controllers
- Notes
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.

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportional</td>
<td>The proportional component depends only on the difference between the setpoint and the process variable</td>
<td>[ u(t) = K_p e(t) ]</td>
</tr>
<tr>
<td>Integral</td>
<td>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.</td>
<td>[ u(t) = K_i \int_0^t e(\tau) d\tau ]</td>
</tr>
<tr>
<td>Derivative</td>
<td>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.</td>
<td>[ u(t) = K_d \frac{de(t)}{dt} ]</td>
</tr>
</tbody>
</table>

Proportional:

Where

\[ e(t) = \text{error value} \]

\[ K_p = \text{proportional gain} \]

\[ K_i = \text{integral gain} \]

\[ K_d = \text{derivative gain} \]

Parallel Linear PID

![Diagram of Parallel Linear PID Controller](image)

Figure 8. Parallel form linear PID controller.
Where

$r = \text{input}$

$y = \text{feedback}$

$e = \text{error}$

$u = \text{output}$

$K_p = \text{proportional gain}$

$K_i = \text{integral gain}$

$K_d = \text{derivative gain}$

**Series Linear PID**

![Series Linear PID Diagram]

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

Where

$e = \text{error}$

$u = \text{output}$

$K_p = \text{proportional gain}$

$K_i = \text{integral gain}$

$K_d = \text{derivative gain}$

**PID equation**

Control law for PID

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$  \hspace{1cm} (43)

Where

$K_p = \text{proportional gain}$
$K_i = \text{integral gain}$

$K_d = \text{derivative gain}$

**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

\[
u(t) = u_{\text{bias}} + K_p e(t) + K_i \int_0^t e(t) \, dt
\] (44)

**Where**

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

$K_p = \text{proportional gain}$

$K_i = \text{integral gain}$

$e(t) = \text{error value}$
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.

![Nonlinear PID input architecture.](image1)

**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, $\alpha$. The tuning parameter determines the degree and direction of the gain shape.

![Nonlinear control law input-output plot.](image2)

**Figure 12.** Nonlinear control law input-output plot.
Table 7. Tuning parameter effect on the gain shape.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha_p$</td>
<td>$&lt; 1$</td>
<td>Smaller gain as error is large, not sensitive to small error.</td>
</tr>
<tr>
<td>$\alpha_p$</td>
<td>$&gt; 1$</td>
<td>Higher gain when error is large, higher gain when error is small and by that more sensitive to small changes.</td>
</tr>
<tr>
<td>$\alpha_i$</td>
<td>$-1 &lt; \alpha_i &lt; 0$</td>
<td>Solves integral windup problem by reducing the integral action when error is large.</td>
</tr>
<tr>
<td>$\alpha_d$</td>
<td>$\alpha_d &gt; 1$</td>
<td>Makes the differential gain small when the error is small which results in less sensitivity to noise.</td>
</tr>
</tbody>
</table>

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

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

(45)

Proportional error expression

$$e_p = e$$

(46)

Integral error expression

$$e_i = edt$$

(47)

Derivative error expression

$$e_d = \frac{d}{dt}e$$

(48)

Reconstructed nonlinear control law

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

(49)

Where

$x$ = input

$y$ = output

$\alpha$ = tuning parameter

$\delta$ = 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.](image)

**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.](image)
Equation

Transfer function of 2P2Z compensator

\[
H_c(s) = K_{DC} \omega_p 0 \left( \frac{s}{w_{z1}} \right) \left( \frac{1 + \frac{s}{w_{p1}}}{1 + \frac{s}{w_{p2}}} \right)
\]  

(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.

![Graph of Phase and Gain Characteristics of a 3P3Z Compensator](image)

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

Equation

Transfer function of 3P3Z compensator

\[
H_c(s) = K_{DC} \omega_p 0 \left( \frac{s}{w_{z1}} \right) \left( \frac{1 + \frac{s}{w_{z2}}}{1 + \frac{s}{w_{z3}}} \right) \left( \frac{1 + \frac{s}{w_{p2}}}{1 + \frac{s}{w_{p3}}} \right)
\]  

(51)
Direct form controllers

DF11

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

![Diagram of DF11](image)

**Figure 16. Representation of DF11.**

**Equations**

General form of discrete time first order transfer function

\[
F(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}
\]  

(52)

Difference equation

\[
u(k) = b_0 e(k) + b_1 e(k - 1) - a_1 u(k - 1)
\]  

(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 \( u(k) \).

![Representation of DF13](image)

**Figure 17. Representation of DF13.**

**Equations**

**General form of a third order transfer function**

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

(54)

**Difference equation**

\[
u(k) = b_0 e(k) + b_1 e(k - 1) + b_2 e(k - 2) + b_3 (k - 3) - a_1 u(k - 1) - a_2 u(k - 2) - a_3 u(k - 3)
\]  

(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.

**Equations**

Transfer function of second order discrete time compensator

\[
F(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}
\]  

(56)

Difference Equation

\[
u(k) = b_0 e(k) + b_1 e(k-1) + b_2 e(k-2) - a_1 u(k-1) - a_2 u(k-2)
\]  

(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

\[ u(k) = b_0 e(k) + b_1 e(k - 1) + b_2 e(k - 2) + b_3 (k - 3) - a_1 u(k - 1) - a_2 u(k - 2) - a_3 u(k - 3) \]  

(58)
ADC definitions
ADC resolution
Quantization error of ADC
Total harmonic distortion (THD)
AC signals
DC signals
Settling time and conversion accuracy
ADC system noise
Notes
ADC definitions

<table>
<thead>
<tr>
<th>Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution = ( n )</td>
<td>The number of bits used to quantify the input</td>
</tr>
<tr>
<td>Number of codes = ( 2^n )</td>
<td>The number of output code combinations</td>
</tr>
<tr>
<td>Full-scale range input = FSR</td>
<td>Sets the converter input range and the LSB voltage</td>
</tr>
<tr>
<td>LSB = FSR / ( 2^n )</td>
<td>The voltage step size of each LSB</td>
</tr>
<tr>
<td>Full-scale input voltage = ((2^n - 1) \cdot 1\text{LSB})</td>
<td>Full-scale input voltage of the analog-to-digital converter (ADC)</td>
</tr>
<tr>
<td>Full-scale output code = (2^n - 1)</td>
<td>Largest code that can be read</td>
</tr>
<tr>
<td>Transfer function: Output Code = FLOOR ( \frac{\text{VIN}}{(\text{FSR}/2^n)} )</td>
<td>Relationship between input voltage and output code</td>
</tr>
</tbody>
</table>

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

\[
\text{FSR} = V_{\text{REF HI}} - V_{\text{REF LO}}
\]  

(59)

One Least Significant Bit

\[
1\text{LSB} = \frac{\text{FSR}}{2^n}
\]  

(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**

\[
1\text{LSB} = \frac{\text{FSR}}{2^n} = \frac{2.5 \text{ V}}{2^{12}} = 610.35 \mu \text{V}
\]

**ADC resolution for differential signals**

![Diagram of ADC full-scale range (FSR) differential.](image)

**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

VREFHI = reference voltage high

VREFLO = reference voltage low
Resolution voltage vs. full-scale range

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

<table>
<thead>
<tr>
<th>Resolution</th>
<th>FSR (Full-Scale Range)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2.5 V</td>
</tr>
<tr>
<td>8</td>
<td>9.67 mV</td>
</tr>
<tr>
<td>10</td>
<td>2.44 mV</td>
</tr>
<tr>
<td>12</td>
<td>610 µV</td>
</tr>
<tr>
<td>14</td>
<td>152.5 µV</td>
</tr>
<tr>
<td>16</td>
<td>38.14 µV</td>
</tr>
<tr>
<td>18</td>
<td>9.53 µV</td>
</tr>
<tr>
<td>20</td>
<td>2.384 µV</td>
</tr>
<tr>
<td>22</td>
<td>596 nV</td>
</tr>
<tr>
<td>24</td>
<td>149 nV</td>
</tr>
</tbody>
</table>

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 $\frac{1}{2}$ 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 $\text{RMSNoise} = \frac{1}{\sqrt{12}} \times \frac{1}{\text{LSB}}$.

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

$$\text{MaxRMSSignal} = \frac{\text{FSR}}{\sqrt{2}} = \frac{1}{\text{LSB}} \times 2^{N-1}$$

$$\text{RMSNoise} = \frac{1}{\sqrt{12}} \times \frac{1}{\text{LSB}}$$

$$\text{SNR} = \frac{\text{MaxRMSSignal}}{\text{RMSNoise}} = \frac{1}{\text{LSB}} \times 2^{N-1} \times \frac{\sqrt{2}}{\sqrt{12}} = 2^{N-1} \sqrt{6}$$
SNR(dB) = 20log(SNR) = [20log(2)]N + 20log(\frac{\sqrt{6}}{2}) \tag{64}

SNR(dB) ≈ 6.02N + 1.76 \tag{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**

\[
SNR = 2^N - 1\sqrt{6} = 2^8 - 1\sqrt{6} = 314
\tag{66}
\]

\[
SNR(dB) = 20\log(314)
\]

\[
SNR(dB) = 6.02(8) + 1.76 = 49.9 \text{ dB}
\]

---

**Total harmonic distortion (THD)**

**Total harmonic distortion (VRMS)**

\[
\text{THD(\%)} = \left(\frac{\text{RMSDistortion}}{\text{MaxRMSSignal}}\right) \times 100 = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \ldots + V_n^2}}{V_1} \times 100 \tag{67}
\]

\[
\text{THD(dB)} = 20\log\left(\frac{\text{RMSDistortion}}{\text{MaxRMSSignal}}\right) \tag{68}
\]

**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
Total harmonic distortion (dBc)

\[
\text{THD(dBc)} = 10 \log \left[ 10\left(\frac{D_2}{D_1}\right) + 10\left(\frac{D_3}{D_1}\right) + 10\left(\frac{D_4}{D_1}\right) + \ldots + 10\left(\frac{D_n}{D_1}\right) \right]
\]

(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
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)

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

(70)

\[
\text{SINAD(dB)} = - 20 \log \left( 10 \left( \frac{-\text{SNR(dB)}}{10} \right) + 10 \left( \frac{-\text{THD(dB)}}{10} \right) \right)
\]

(71)

Effective Number of Bits (ENOB)

\[
\text{ENOB} = \frac{\text{SINAD(dB)} - 1.76 \text{dB}}{6.02}
\]

(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 RMS Distortion = 50 \mu V_{RMS}

RMS Noise = 100 \mu V_{RMS}

Answer

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

\[
\text{THD(dB)} = 20 \log \left( \frac{50 \text{\mu V}_{\text{RMS}}}{100 \text{\mu V}_{\text{RMS}}} \right) = - 90.9 \text{ dB}
\]
\[
\text{SINAD(dB)} = 20 \log \left( \frac{1.76 \ \mu V_{\text{RMS}}}{\sqrt{(100 \ \mu V_{\text{RMS}})^2 + (50 \ \mu V_{\text{RMS}})^2}} \right) = 83.9 \ \text{dB}
\]

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

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

**DC signals**

\[
\text{NoiseFreeResolution} = \log_2 \left( \frac{2^N}{\text{PeaktoPeakNoiseinLSB}} \right)
\]

\[
\text{EffectiveResolution} = \log_2 \left( \frac{2^N}{\text{rmsNoiseLSB}} \right)
\]

\[
\text{PeaktoPeakNoiseinLSB} \approx 6.6 \times \text{rmsNoiseLSB}
\]

\[
\text{EffectiveResolution} \approx \text{NoiseFreeResolution} + 2.7
\]

**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**

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

\[
\text{EffectiveResolution} = \log_2 \left( \frac{2^{24}}{5.6} \right) = 23.9
\]

\[
\text{EffectiveResolution} = 21.2 + 2.7 = 23.9
\]
Settling time and conversion accuracy

Figure 26. Settling time and conversion accuracy.

Table 9. Conversion accuracy achieved after a specified time.

<table>
<thead>
<tr>
<th>N&lt;sub&gt;TC&lt;/sub&gt;</th>
<th>Accuracy in Bits (N)</th>
<th>N&lt;sub&gt;TC&lt;/sub&gt;</th>
<th>Accuracy in Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.44</td>
<td>10</td>
<td>14.43</td>
</tr>
<tr>
<td>2</td>
<td>2.89</td>
<td>11</td>
<td>15.87</td>
</tr>
<tr>
<td>3</td>
<td>4.33</td>
<td>12</td>
<td>17.31</td>
</tr>
<tr>
<td>4</td>
<td>5.77</td>
<td>13</td>
<td>18.76</td>
</tr>
<tr>
<td>5</td>
<td>7.21</td>
<td>14</td>
<td>20.20</td>
</tr>
<tr>
<td>6</td>
<td>8.66</td>
<td>15</td>
<td>21.64</td>
</tr>
<tr>
<td>7</td>
<td>10.10</td>
<td>16</td>
<td>23.08</td>
</tr>
<tr>
<td>8</td>
<td>11.54</td>
<td>17</td>
<td>24.53</td>
</tr>
<tr>
<td>9</td>
<td>12.98</td>
<td>18</td>
<td>25.97</td>
</tr>
</tbody>
</table>

Where

N = the number of bits of accuracy the RC circuit has settled to after N<sub>TC</sub> number of time constants

N<sub>TC</sub> = 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<sub>REF</sub>.

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

<table>
<thead>
<tr>
<th>N&lt;sub&gt;TC&lt;/sub&gt;</th>
<th>Accuracy in Bits (N)</th>
<th>N&lt;sub&gt;TC&lt;/sub&gt;</th>
<th>Accuracy in Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>5.5</td>
<td>17</td>
<td>11.78</td>
</tr>
<tr>
<td>9</td>
<td>6.24</td>
<td>18</td>
<td>12.48</td>
</tr>
<tr>
<td>10</td>
<td>6.93</td>
<td>19</td>
<td>13.17</td>
</tr>
<tr>
<td>11</td>
<td>7.62</td>
<td>20</td>
<td>13.86</td>
</tr>
<tr>
<td>12</td>
<td>8.32</td>
<td>21</td>
<td>14.56</td>
</tr>
<tr>
<td>13</td>
<td>9.01</td>
<td>22</td>
<td>15.25</td>
</tr>
<tr>
<td>14</td>
<td>9.70</td>
<td>23</td>
<td>15.94</td>
</tr>
<tr>
<td>15</td>
<td>10.40</td>
<td>24</td>
<td>16.64</td>
</tr>
<tr>
<td>16</td>
<td>11.04</td>
<td>25</td>
<td>17.33</td>
</tr>
</tbody>
</table>
Where

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

\( N \) = 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**

\[ V_{FSR_{RMS}} = \frac{V_{FSR} \times 0.707}{2} \]  \hspace{1cm} (77)

\[ \text{SNR}_{ADC} = 20 \times \log \left( \frac{V_{FSR_{RMS}}}{V_{nADC}} \right) \]  \hspace{1cm} (78)

\[ V_{nADC} = \frac{V_{FSR_{RMS}}}{10 \left( \frac{\text{SNR}_{ADC}}{20} \right)} \]  \hspace{1cm} (79)

\[ V_{nT} = \sqrt{(V_{nADC})^2 + (V_{nAmp})^2 + (V_{nRef})^2} \]  \hspace{1cm} (80)

**Where**

\( V_{FSR} \) = 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.

\( \text{SNR}_{ADC} \) = 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
Comparator

- Basic operation
- Offset and hysteresis
- Propagation delay
- Notes

Real-Time Control Reference Guide

October 2021
Basic operation

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

![Comparator diagram](image)

**Figure 28. Comparator diagram.**

**Table 11. Comparator inputs to output.**

<table>
<thead>
<tr>
<th>Inputs Condition</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>IN+ &gt; IN-</td>
<td>HIGH (V+)</td>
</tr>
<tr>
<td>IN+ = IN-</td>
<td>Indeterminate (chatters - see Hysteresis)</td>
</tr>
<tr>
<td>IN+ &lt; IN-</td>
<td>LOW (V-)</td>
</tr>
</tbody>
</table>

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: $V_{TH}$, $V_{OFF}$, $V_{HYST}$.

![Comparator hysteresis](image)

**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.

![Diagram of comparator input to output switching delay](image)

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

**Table 12. Primitive data types.**

<table>
<thead>
<tr>
<th>Type</th>
<th>Default</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>False</td>
<td>1 bit</td>
</tr>
<tr>
<td>Byte</td>
<td>0</td>
<td>8 bits</td>
</tr>
<tr>
<td>Char</td>
<td>\u0000</td>
<td>16 bits</td>
</tr>
<tr>
<td>Short</td>
<td>0</td>
<td>16 bits</td>
</tr>
<tr>
<td>Int</td>
<td>0</td>
<td>32 bits</td>
</tr>
<tr>
<td>Long</td>
<td>0</td>
<td>64 bits</td>
</tr>
<tr>
<td>Float</td>
<td>0.0f</td>
<td>32 bits</td>
</tr>
<tr>
<td>Double</td>
<td>0.0d</td>
<td>64 bits</td>
</tr>
</tbody>
</table>

**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.**

<table>
<thead>
<tr>
<th>Multiplier</th>
<th>Prefix</th>
<th>Abbreviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(10^9)</td>
<td>Giga</td>
<td>G</td>
</tr>
<tr>
<td>(10^6)</td>
<td>Mega</td>
<td>M</td>
</tr>
<tr>
<td>(10^3)</td>
<td>Kilo</td>
<td>k</td>
</tr>
<tr>
<td>(10^{-3})</td>
<td>Milli</td>
<td>m</td>
</tr>
<tr>
<td>(10^{-6})</td>
<td>Micro</td>
<td>u</td>
</tr>
<tr>
<td>(10^{-9})</td>
<td>Nano</td>
<td>n</td>
</tr>
<tr>
<td>(10^{-12})</td>
<td>Pico</td>
<td>p</td>
</tr>
<tr>
<td>(10^{-15})</td>
<td>Femto</td>
<td>f</td>
</tr>
</tbody>
</table>

**Table 14. ASCII table.**

<table>
<thead>
<tr>
<th>Binary</th>
<th>Oct</th>
<th>Dec</th>
<th>Hex</th>
<th>Chr</th>
</tr>
</thead>
<tbody>
<tr>
<td>0100000</td>
<td>040</td>
<td>32</td>
<td>20</td>
<td>Space</td>
</tr>
<tr>
<td>0100001</td>
<td>041</td>
<td>33</td>
<td>21</td>
<td>!</td>
</tr>
<tr>
<td>0100010</td>
<td>042</td>
<td>34</td>
<td>22</td>
<td>*</td>
</tr>
<tr>
<td>0100011</td>
<td>043</td>
<td>35</td>
<td>23</td>
<td>#</td>
</tr>
<tr>
<td>0100100</td>
<td>044</td>
<td>36</td>
<td>24</td>
<td>$</td>
</tr>
<tr>
<td>0100101</td>
<td>045</td>
<td>37</td>
<td>25</td>
<td>%</td>
</tr>
<tr>
<td>0100110</td>
<td>046</td>
<td>38</td>
<td>26</td>
<td>&amp;</td>
</tr>
<tr>
<td>0100111</td>
<td>047</td>
<td>39</td>
<td>27</td>
<td>'</td>
</tr>
<tr>
<td>0101000</td>
<td>050</td>
<td>40</td>
<td>28</td>
<td>(</td>
</tr>
<tr>
<td>Binary</td>
<td>Oct</td>
<td>Dec</td>
<td>Hex</td>
<td>Chr</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>0101001</td>
<td>051</td>
<td>41</td>
<td>29</td>
<td>)</td>
</tr>
<tr>
<td>0101010</td>
<td>052</td>
<td>42</td>
<td>2A</td>
<td>*</td>
</tr>
<tr>
<td>0101011</td>
<td>053</td>
<td>43</td>
<td>2B</td>
<td>+</td>
</tr>
<tr>
<td>0101100</td>
<td>054</td>
<td>44</td>
<td>2C</td>
<td>.</td>
</tr>
<tr>
<td>0101101</td>
<td>055</td>
<td>45</td>
<td>2D</td>
<td>-</td>
</tr>
<tr>
<td>0101110</td>
<td>056</td>
<td>46</td>
<td>2E</td>
<td>.</td>
</tr>
<tr>
<td>0101111</td>
<td>057</td>
<td>47</td>
<td>2F</td>
<td>/</td>
</tr>
<tr>
<td>0110000</td>
<td>060</td>
<td>48</td>
<td>30</td>
<td>0</td>
</tr>
<tr>
<td>0110001</td>
<td>061</td>
<td>49</td>
<td>31</td>
<td>1</td>
</tr>
<tr>
<td>0110010</td>
<td>062</td>
<td>50</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>0110011</td>
<td>063</td>
<td>51</td>
<td>33</td>
<td>3</td>
</tr>
<tr>
<td>0110100</td>
<td>064</td>
<td>52</td>
<td>34</td>
<td>4</td>
</tr>
<tr>
<td>0110101</td>
<td>065</td>
<td>53</td>
<td>35</td>
<td>5</td>
</tr>
<tr>
<td>0110110</td>
<td>066</td>
<td>54</td>
<td>36</td>
<td>6</td>
</tr>
<tr>
<td>0110111</td>
<td>067</td>
<td>55</td>
<td>37</td>
<td>7</td>
</tr>
<tr>
<td>0111000</td>
<td>070</td>
<td>56</td>
<td>38</td>
<td>8</td>
</tr>
<tr>
<td>0111001</td>
<td>071</td>
<td>57</td>
<td>39</td>
<td>9</td>
</tr>
<tr>
<td>0111010</td>
<td>072</td>
<td>58</td>
<td>3A</td>
<td>:</td>
</tr>
<tr>
<td>0111011</td>
<td>073</td>
<td>59</td>
<td>3B</td>
<td>;</td>
</tr>
<tr>
<td>0111100</td>
<td>074</td>
<td>60</td>
<td>3C</td>
<td>&lt;</td>
</tr>
<tr>
<td>0111101</td>
<td>075</td>
<td>61</td>
<td>3D</td>
<td>=</td>
</tr>
<tr>
<td>0111110</td>
<td>076</td>
<td>62</td>
<td>3E</td>
<td>&gt;</td>
</tr>
<tr>
<td>0111111</td>
<td>077</td>
<td>63</td>
<td>3F</td>
<td>?</td>
</tr>
<tr>
<td>1000000</td>
<td>100</td>
<td>64</td>
<td>40</td>
<td>@</td>
</tr>
<tr>
<td>1000001</td>
<td>101</td>
<td>65</td>
<td>41</td>
<td>A</td>
</tr>
<tr>
<td>1000010</td>
<td>102</td>
<td>66</td>
<td>42</td>
<td>B</td>
</tr>
<tr>
<td>1000011</td>
<td>103</td>
<td>67</td>
<td>43</td>
<td>C</td>
</tr>
<tr>
<td>1000100</td>
<td>104</td>
<td>68</td>
<td>44</td>
<td>D</td>
</tr>
<tr>
<td>1000101</td>
<td>105</td>
<td>69</td>
<td>45</td>
<td>E</td>
</tr>
<tr>
<td>1000110</td>
<td>106</td>
<td>70</td>
<td>46</td>
<td>F</td>
</tr>
<tr>
<td>1000111</td>
<td>107</td>
<td>71</td>
<td>47</td>
<td>G</td>
</tr>
<tr>
<td>1001000</td>
<td>110</td>
<td>72</td>
<td>48</td>
<td>H</td>
</tr>
<tr>
<td>1001001</td>
<td>111</td>
<td>73</td>
<td>49</td>
<td>I</td>
</tr>
<tr>
<td>1001010</td>
<td>112</td>
<td>74</td>
<td>4A</td>
<td>J</td>
</tr>
<tr>
<td>1001011</td>
<td>113</td>
<td>75</td>
<td>4B</td>
<td>K</td>
</tr>
<tr>
<td>1001100</td>
<td>114</td>
<td>76</td>
<td>4C</td>
<td>L</td>
</tr>
<tr>
<td>1001101</td>
<td>115</td>
<td>77</td>
<td>4D</td>
<td>M</td>
</tr>
<tr>
<td>1001110</td>
<td>116</td>
<td>78</td>
<td>4E</td>
<td>N</td>
</tr>
<tr>
<td>1001111</td>
<td>117</td>
<td>79</td>
<td>4F</td>
<td>O</td>
</tr>
<tr>
<td>1010000</td>
<td>120</td>
<td>80</td>
<td>50</td>
<td>P</td>
</tr>
<tr>
<td>1010001</td>
<td>121</td>
<td>81</td>
<td>51</td>
<td>Q</td>
</tr>
<tr>
<td>1010010</td>
<td>122</td>
<td>82</td>
<td>52</td>
<td>R</td>
</tr>
<tr>
<td>1010011</td>
<td>123</td>
<td>83</td>
<td>53</td>
<td>S</td>
</tr>
<tr>
<td>1010100</td>
<td>124</td>
<td>84</td>
<td>54</td>
<td>T</td>
</tr>
<tr>
<td>1010101</td>
<td>125</td>
<td>85</td>
<td>55</td>
<td>U</td>
</tr>
<tr>
<td>1010110</td>
<td>126</td>
<td>86</td>
<td>56</td>
<td>V</td>
</tr>
<tr>
<td>1010111</td>
<td>127</td>
<td>87</td>
<td>57</td>
<td>W</td>
</tr>
<tr>
<td>Binary</td>
<td>Oct</td>
<td>Dec</td>
<td>Hex</td>
<td>Chr</td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>1011000</td>
<td>130</td>
<td>88</td>
<td>58</td>
<td>X</td>
</tr>
<tr>
<td>1011001</td>
<td>131</td>
<td>89</td>
<td>59</td>
<td>Y</td>
</tr>
<tr>
<td>1011010</td>
<td>132</td>
<td>90</td>
<td>5A</td>
<td>Z</td>
</tr>
<tr>
<td>1011011</td>
<td>133</td>
<td>91</td>
<td>5B</td>
<td>]</td>
</tr>
<tr>
<td>1011100</td>
<td>134</td>
<td>92</td>
<td>5C</td>
<td>\</td>
</tr>
<tr>
<td>1011101</td>
<td>135</td>
<td>93</td>
<td>5D</td>
<td>}</td>
</tr>
<tr>
<td>1011110</td>
<td>136</td>
<td>94</td>
<td>5E</td>
<td>^</td>
</tr>
<tr>
<td>1011111</td>
<td>137</td>
<td>95</td>
<td>5F</td>
<td>_</td>
</tr>
<tr>
<td>1100000</td>
<td>140</td>
<td>96</td>
<td>60</td>
<td>'</td>
</tr>
<tr>
<td>1100001</td>
<td>141</td>
<td>97</td>
<td>61</td>
<td>a</td>
</tr>
<tr>
<td>1100010</td>
<td>142</td>
<td>98</td>
<td>62</td>
<td>b</td>
</tr>
<tr>
<td>1100011</td>
<td>143</td>
<td>99</td>
<td>63</td>
<td>c</td>
</tr>
<tr>
<td>1100100</td>
<td>144</td>
<td>100</td>
<td>64</td>
<td>d</td>
</tr>
<tr>
<td>1100101</td>
<td>145</td>
<td>101</td>
<td>65</td>
<td>e</td>
</tr>
<tr>
<td>1100110</td>
<td>146</td>
<td>102</td>
<td>66</td>
<td>f</td>
</tr>
<tr>
<td>1100111</td>
<td>147</td>
<td>103</td>
<td>67</td>
<td>g</td>
</tr>
<tr>
<td>1101000</td>
<td>150</td>
<td>104</td>
<td>68</td>
<td>h</td>
</tr>
<tr>
<td>1101001</td>
<td>151</td>
<td>105</td>
<td>69</td>
<td>i</td>
</tr>
<tr>
<td>1101010</td>
<td>152</td>
<td>106</td>
<td>6A</td>
<td>j</td>
</tr>
<tr>
<td>1101011</td>
<td>153</td>
<td>107</td>
<td>6B</td>
<td>k</td>
</tr>
<tr>
<td>1101100</td>
<td>154</td>
<td>108</td>
<td>6C</td>
<td>l</td>
</tr>
<tr>
<td>1101101</td>
<td>155</td>
<td>109</td>
<td>6D</td>
<td>m</td>
</tr>
<tr>
<td>1101110</td>
<td>156</td>
<td>110</td>
<td>6E</td>
<td>n</td>
</tr>
<tr>
<td>1101111</td>
<td>157</td>
<td>111</td>
<td>6F</td>
<td>o</td>
</tr>
<tr>
<td>1110000</td>
<td>160</td>
<td>112</td>
<td>70</td>
<td>p</td>
</tr>
<tr>
<td>1110001</td>
<td>161</td>
<td>113</td>
<td>71</td>
<td>q</td>
</tr>
<tr>
<td>1110010</td>
<td>162</td>
<td>114</td>
<td>72</td>
<td>r</td>
</tr>
<tr>
<td>1110011</td>
<td>163</td>
<td>115</td>
<td>73</td>
<td>s</td>
</tr>
<tr>
<td>1110100</td>
<td>164</td>
<td>116</td>
<td>74</td>
<td>t</td>
</tr>
<tr>
<td>1110101</td>
<td>165</td>
<td>117</td>
<td>75</td>
<td>u</td>
</tr>
<tr>
<td>1110110</td>
<td>166</td>
<td>118</td>
<td>76</td>
<td>v</td>
</tr>
<tr>
<td>1110111</td>
<td>167</td>
<td>119</td>
<td>77</td>
<td>w</td>
</tr>
<tr>
<td>1111000</td>
<td>170</td>
<td>120</td>
<td>78</td>
<td>x</td>
</tr>
<tr>
<td>1111001</td>
<td>171</td>
<td>121</td>
<td>79</td>
<td>y</td>
</tr>
<tr>
<td>1111010</td>
<td>172</td>
<td>122</td>
<td>7A</td>
<td>z</td>
</tr>
<tr>
<td>1111011</td>
<td>173</td>
<td>123</td>
<td>7B</td>
<td>(</td>
</tr>
<tr>
<td>1111100</td>
<td>174</td>
<td>124</td>
<td>7C</td>
<td>)</td>
</tr>
<tr>
<td>1111101</td>
<td>175</td>
<td>125</td>
<td>7D</td>
<td></td>
</tr>
<tr>
<td>1111110</td>
<td>176</td>
<td>126</td>
<td>7E</td>
<td>-</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast at performing math operations</td>
<td>Math operations: MUL, ADD, SUB</td>
</tr>
<tr>
<td>Fast at trigonometric operations</td>
<td>Trigonometric operations: SIN, COS, ATAN, DIV</td>
</tr>
<tr>
<td>Fast at saturation operations</td>
<td>Saturation operations check for out of bounds conditions and if detected the value is clamped or saturated</td>
</tr>
<tr>
<td>Contains deterministic execution</td>
<td>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</td>
</tr>
<tr>
<td>Characteristic</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Fast access to peripheral registers</td>
<td>For example, reading sensor inputs (example: ADC) and writing to output registers (example: EPWM)</td>
</tr>
<tr>
<td>Fast Interrupt Response</td>
<td>The processor needs to respond to periodic interrupt events with low latency to satisfy real time deadlines</td>
</tr>
<tr>
<td>Floating-point operation capability</td>
<td>Floating-point operations are generally easier to use which make the control algorithms more robust</td>
</tr>
<tr>
<td>32-bit and 64-bit data precision</td>
<td>A processor that can be scaled in precision based on the application is ideal</td>
</tr>
<tr>
<td>Multicore Support</td>
<td>Having dual cores or more helps increase the amount of tasks that are executed given a specified amount of time</td>
</tr>
</tbody>
</table>

**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.

![Signal chain diagram](image)

**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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLASH</td>
<td>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.</td>
</tr>
<tr>
<td>Random Access Memory (RAM)</td>
<td>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.</td>
</tr>
<tr>
<td>Read Only Memory (ROM)</td>
<td>ROM contains specialized tasks that will never change.</td>
</tr>
<tr>
<td>CACHE</td>
<td>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 processor 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.</td>
</tr>
<tr>
<td>Tightly Coupled Memory (TCM)</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.](image-url)
**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.

![ Interrupt Flow Diagram](image)

*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.
Encoders

- Encoder definitions
- Types of encoders
- Description of encoders
- Absolute Vs incremental encoders
- Notes
Encoder definitions

<table>
<thead>
<tr>
<th>Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
<td>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.</td>
</tr>
<tr>
<td>Speed</td>
<td>Maximum speed at which the device can rotate without sustaining physical damage, often defined as revolutions per minute (RPM).</td>
</tr>
<tr>
<td>Positional Accuracy</td>
<td>It is the maximum error of a reading, in arc seconds. One degree is 3600 arc seconds.</td>
</tr>
<tr>
<td>Absolute Encoder Codes</td>
<td>Binary, BCD or Gray format.</td>
</tr>
<tr>
<td>Incremental Encoder Signal Types</td>
<td>Quadrature - Refers to A and B channels that are 90° out of phase. Single Channel - Unidirectional allowing one count per physical line. Pulse &amp; Direction - Direction channel may be combined with single-channel counting or multiple-channel counting with quadrature.</td>
</tr>
<tr>
<td>Encoder output</td>
<td>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.</td>
</tr>
</tbody>
</table>

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.
Pulse width modulation (PWM)

PWM definitions
- Duty cycle
- Resolution
- Deadband
- Notes
PWM definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period (T)</td>
<td>The duration after which the PWM pattern will repeat itself</td>
</tr>
<tr>
<td>Frequency (F)</td>
<td>Inverse of the period</td>
</tr>
<tr>
<td>On-Time</td>
<td>The amount of time the output signal has a logic high value during a period</td>
</tr>
<tr>
<td>Off-Time</td>
<td>The amount of time the output signal has a logic low value during a period</td>
</tr>
<tr>
<td>Amplitude</td>
<td>The max voltage of the output signal</td>
</tr>
</tbody>
</table>

\[
\text{Duty Cycle (\%)} = \frac{\text{"ON Time"}}{\text{Period}} \times 100 \tag{81}
\]

\[
\text{Average Output Voltage} \left(\bar{V}\right) = \frac{\text{"ON Time"}}{\text{Period}} \times \text{Amplitude} \tag{82}
\]
**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**

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

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

**Resolution**

\[
\text{PWM resolution (\%)} = \frac{F_{PWM}}{F_{PWMCLK}} \times 100\% \quad (85)
\]

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

**Where**

\( \text{PWM}_{\text{resolution}} \) = 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.

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

Example

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

Answer

\[
\text{PWM resolution (\%)} = \frac{f_{\text{PWM}}}{f_{\text{PWMCLK}}} \times 100\% = \frac{1}{\frac{500\text{ns}}{200\text{MHz}}} \times 100 = 1\%
\]

\[
\text{PWM resolution (bits)} = \log_2\left(\frac{T_{\text{PWM}}}{T_{\text{PWMCLK}}}\right) = \log_2\left(\frac{\frac{500\text{ns}}{1\text{MHz}}}{1\text{MHz}}\right) = 6.64 \text{ bits} \rightarrow 6 \text{ bits}
\]

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.
**DAC definitions**

<table>
<thead>
<tr>
<th>Description</th>
<th>Formula</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution = $n$</td>
<td>The number of bits used to represent each input code</td>
<td></td>
</tr>
<tr>
<td>Number of codes = $2^n$</td>
<td>The number of possible codes that can be enumerated</td>
<td></td>
</tr>
<tr>
<td>Full-scale range output = FSR</td>
<td>The limit of the output voltage range</td>
<td></td>
</tr>
<tr>
<td>$\text{LSB} = \frac{\text{FSR}}{2^n}$</td>
<td>The voltage difference between two consecutive codes</td>
<td></td>
</tr>
<tr>
<td>Full-scale output voltage = $2^n - 1 \times \text{1LSB}$</td>
<td>The maximum voltage that can be output</td>
<td></td>
</tr>
<tr>
<td>Full-scale input code = $2^n - 1$</td>
<td>The largest code that can be represented</td>
<td></td>
</tr>
<tr>
<td>Transfer function: $V_{\text{Out}} = \frac{\text{FSR}}{2^n} \times \text{Code}$</td>
<td>Relationship between input code and output voltage</td>
<td></td>
</tr>
</tbody>
</table>

**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:

$$V_{\text{Out}} = \left(\frac{\text{FSR}}{2^n} \times \text{Code}\right)$$  \hspace{1cm} (87)

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

$$V_{\text{Out}} = \left(\frac{\text{FSR}}{2^n} \times \text{Code}\right) + E_{\text{Offset}}$$  \hspace{1cm} (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:

\[ V_{\text{Out}} = \left( \frac{\text{FSR}}{2^n} \right) \cdot \text{Code} \]  

(89)

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

\[ V_{\text{Out}} = \left( \frac{\text{FSR}}{2^n} \right) \cdot \text{Code} \cdot E_{\text{Gain}} \]  

(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.

Gain

Measured

Gain

Error

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 zero-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.
**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 44. DAC DNL.**

**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.

<table>
<thead>
<tr>
<th>Error</th>
<th>Offset</th>
<th>Gain</th>
<th>Zero-Code</th>
<th>Full-Scale</th>
<th>DNL</th>
<th>INL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offset</td>
<td>-</td>
<td>-</td>
<td>Correlated</td>
<td>Correlated</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Gain</td>
<td>Correlated</td>
<td>-</td>
<td>-</td>
<td>Correlated</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zero-Code</td>
<td>Correlated</td>
<td>-</td>
<td>Correlated</td>
<td>-</td>
<td>-</td>
<td>Correlated</td>
</tr>
<tr>
<td>Full-Scale</td>
<td>Correlated</td>
<td>Correlated</td>
<td>-</td>
<td>-</td>
<td>Correlated</td>
<td>Correlated</td>
</tr>
<tr>
<td>DNL</td>
<td>-</td>
<td>-</td>
<td>Correlated</td>
<td>-</td>
<td>Correlated</td>
<td>-</td>
</tr>
<tr>
<td>INL</td>
<td>-</td>
<td>-</td>
<td>Correlated</td>
<td>Correlated</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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

\[
\text{TUE} = \sqrt{E_{\text{Offset}}^2 + E_{\text{Gain}}^2 + E_{\text{INL}}^2}
\]  

(91)

Where

\(E_{\text{Offset}}\) = The static component of output error across the transfer function. See DAC Offset Error.

\(E_{\text{Gain}}\) = The proportionate component of output error across the transfer function. See DAC Gain Error.

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

Table 19. Unit conversions for error.

<table>
<thead>
<tr>
<th>Convert</th>
<th>Codes</th>
<th>Volts</th>
<th>%</th>
<th>ppm</th>
</tr>
</thead>
<tbody>
<tr>
<td>From</td>
<td>Codes</td>
<td>Volts • 2^n / V_{FSR}</td>
<td>% • V_{FSR} / 100</td>
<td>ppm • 10^n / 10^6</td>
</tr>
<tr>
<td></td>
<td>Codes • V_{FSR} / 2^n</td>
<td>Volts • 100 / V_{FSR}</td>
<td>% • 10^6 / 100</td>
<td>-</td>
</tr>
<tr>
<td>To</td>
<td>Codes • 10^n / 2^n</td>
<td>Volts • 10^6 / V_{FSR}</td>
<td>-</td>
<td>ppm • 10^6 / 10^6</td>
</tr>
</tbody>
</table>

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 \(E_{\text{Gain}}\) as a uniform contributor of error across the transfer function, but \(E_{\text{Gain}}\) 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.

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.

<table>
<thead>
<tr>
<th>End Points</th>
<th>Ideal Line</th>
<th>End Point Corrected Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min Code</td>
<td>Code = 0</td>
<td>Minimum Linear Code</td>
</tr>
<tr>
<td>Min Voltage</td>
<td>Ideal Voltage</td>
<td>Measured Voltage</td>
</tr>
<tr>
<td>Max Code</td>
<td>Code = 2^n-1</td>
<td>Maximum Linear Code</td>
</tr>
<tr>
<td>Max Voltage</td>
<td>Ideal Voltage</td>
<td>Measured Voltage</td>
</tr>
</tbody>
</table>

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).
DAC settling time

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

**Figure 48. DAC load regulation.**
Mathematical models

- Laplace transforms
- Transfer function
- Transient response
- Frequency response
- Z-domain
- Notes
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 transform $f(s)$ is,

$$f(s) = L[f(t)] = \int_0^\infty f(t)e^{-st}dt$$

**Table 21. Important Laplace transform pairs.**

<table>
<thead>
<tr>
<th>$f(t)$</th>
<th>$f(s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step function, $u(t)$</td>
<td>$\frac{1}{s}$</td>
</tr>
<tr>
<td>$e^{-at}$</td>
<td>$\frac{1}{s+a}$</td>
</tr>
<tr>
<td>$\sin(\omega t)$</td>
<td>$\frac{\omega}{s^2 + \omega^2}$</td>
</tr>
<tr>
<td>$\cos(\omega t)$</td>
<td>$\frac{s}{s^2 + \omega^2}$</td>
</tr>
<tr>
<td>$t^n$</td>
<td>$\frac{n!}{s^{n+1}}$</td>
</tr>
<tr>
<td>$f^{(k)}(t) = \frac{d^k f(t)}{dt^k}$</td>
<td>$s^k F(s) - s^{k-1} f(0') - s^{k-2} f''(0') - ... - s f^{(k-1)}(0')$</td>
</tr>
<tr>
<td>$\int_{-\infty}^{t} f(\tau) d\tau$</td>
<td>$\frac{F(s)}{s} + \frac{1}{s} \int_{-\infty}^{0} f(\tau) d\tau$</td>
</tr>
<tr>
<td>Impulse function $\delta(t)$</td>
<td>1</td>
</tr>
<tr>
<td>$e^{-at}\sin(\omega t)$</td>
<td>$\frac{\omega}{(s+a)^2 + \omega^2}$</td>
</tr>
<tr>
<td>$e^{-at}\cos(\omega t)$</td>
<td>$\frac{s + a}{(s+a)^2 + \omega^2}$</td>
</tr>
<tr>
<td>$\frac{1}{\omega} \left[ (\alpha - \omega)^2 + \omega^2 \right]^{1/2} e^{-at}\sin(\omega t + \phi)$</td>
<td>$\phi = \tan^{-1} \left( -\frac{\omega}{\alpha} \right)$</td>
</tr>
<tr>
<td>$\frac{\omega}{\sqrt{1 - \zeta^2}} e^{\omega \sqrt{1 - \zeta^2} t}$, $\zeta &lt; 1$</td>
<td>$\phi = \tan^{-1} \left( -\frac{\omega}{\alpha} \right)$</td>
</tr>
<tr>
<td>$\frac{\omega}{\sqrt{1 + \zeta^2}} e^{-\omega \sqrt{1 + \zeta^2} t}$, $\zeta &gt; 1$</td>
<td>$\phi = \tan^{-1} \left( -\frac{\omega}{\alpha} \right)$</td>
</tr>
</tbody>
</table>

**Transfer function**

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

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

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

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

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

$$\beta(s) = b_m s^m + \ldots + b_1 s + b_0 \quad (93)$$

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

$$\alpha(s) = a_n s^n + \ldots + a_1 s + a_0 \quad (94)$$

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

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

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

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

**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,

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

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

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

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

$$y(t) = \varepsilon_1 e^{r_1 t} + \varepsilon_2 e^{r_2 t} + \ldots + \varepsilon_{n+1} e^{r_{n+1} t} + \ldots + \varepsilon_q e^{r_q t} \quad (99)$$

**Table 22. Response portions.**

<table>
<thead>
<tr>
<th>Response Type</th>
<th>Representation</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transient Response</td>
<td>$y_c(t)$</td>
<td>$\varepsilon_1 e^{r_1 t} + \varepsilon_2 e^{r_2 t} + \ldots + \varepsilon_n e^{r_n t}$</td>
<td>The $n$ terms in $y(t)$ with roots originating from comprise the transient response</td>
</tr>
<tr>
<td>Steady State Response</td>
<td>$y_p(t)$</td>
<td>$\varepsilon_n + 1 e^{r_n t} + 1^t + \ldots + \varepsilon_q e^{r_q t}$</td>
<td>The $a-n$ terms originating from $u(s)$ comprise the steady state response</td>
</tr>
</tbody>
</table>

**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 $\Phi = \beta - \alpha$ 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

\[ \text{Voltage gain (dB)} = 20 \log \left( \frac{V_{\text{OUT}}}{V_{\text{IN}}} \right) \]  

(100)

Power gain in decibels

\[ \text{Power gain (dB)} = 10 \log \left( \frac{P_{\text{OUT}}}{P_{\text{IN}}} \right) \]  

(101)

Used for input or output power

\[ \text{Power measured (dBm)} = 10 \log \left( \frac{\text{Power measured (W)}}{1 \text{ mW}} \right) \]  

(102)

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

<table>
<thead>
<tr>
<th>A (V/V)</th>
<th>A (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>-60</td>
</tr>
<tr>
<td>0.01</td>
<td>-40</td>
</tr>
<tr>
<td>0.1</td>
<td>-20</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>100</td>
<td>40</td>
</tr>
<tr>
<td>1,000</td>
<td>60</td>
</tr>
<tr>
<td>10,000</td>
<td>80</td>
</tr>
<tr>
<td>100,000</td>
<td>100</td>
</tr>
<tr>
<td>1,000,000</td>
<td>120</td>
</tr>
<tr>
<td>10,000,000</td>
<td>140</td>
</tr>
</tbody>
</table>

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

Magnitude \( (f = f_p) = -3 \, \text{dB} \)

Magnitude \( (f > f_p) = -20 \, \text{dB/decade} \)

Phase \( (f = f_p) = -45^\circ \)

Phase \( (0.1 \, f_p < f < 10 \, f_p) = -45^\circ/\text{decade} \)

Phase \( (f > 10 \, f_p) = -90^\circ \)

Phase \( (f < 0.1 \, f_p) = 0^\circ \)

Pole (equations)

As a complex number

\[ G_V = \frac{V_{OUT}}{V_{IN}} = \frac{G_{DC}}{f f_p + 1} \]

(103)
Magnitude

\[ G_V = \frac{V_{OUT}}{V_{IN}} = \frac{G_{DC}}{\sqrt{(\frac{f}{f_p})^2 + 1}} \]  

(104)

Phase Shift

\[ \theta = -\tan^{-1}\left(\frac{f}{f_p}\right) \]  

(105)

Magnitude in dB

\[ G_{dB} = 20\log(G_V) \]  

(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  
\( \theta \) = 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 \( (f < f_z) = 0 \text{ dB} \)

Magnitude \( (f = f_z) = +3 \text{ dB} \)

Magnitude \( (f > f_z) = +20 \text{ dB/decade} \)

Phase \( (f = f_z) = +45^\circ \)

Phase \( (0.1 f_z < f < 10 f_z) = +45^\circ/\text{decade} \)

Phase \( (f > 10 f_z) = +90^\circ \)

Phase \( (f < 0.1 f_z) = 0^\circ \)

Zero (equations)

As a complex number

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

(107)

Magnitude
\[ G_V = \frac{V_{OUT}}{V_{IN}} = G_{DC} \sqrt{\left(\frac{f}{f_z}\right)^2 + 1} \]  \hspace{1cm} (108)

Phase Shift
\[ \theta = \tan^{-1}\left(\frac{f}{f_z}\right) \]  \hspace{1cm} (109)

Magnitude in dB
\[ G_{dB} = 20 \log(G_V) \]  \hspace{1cm} (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  
\( \theta \) = phase shift of the signal from input to output  
\( j \) = indicates imaginary number or \( \sqrt{-1} \)

**Z-domain**

Definition of Bilateral Z-Transform
\[ x(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n} \]  \hspace{1cm} (111)

Definition of Unilateral Z-Transform
\[ x(z) = \sum_{n=0}^{\infty} x[n]z^{-n} \]  \hspace{1cm} (112)

**Where:**

\( x[n] \) = discrete time signal  
\( x(z) \) = the z-domain transform of the discrete time signal  
\( n \) = integer  
\( z \) = complex number (\( Ae^{j\phi} = A(\cos \phi + jsin \phi) \))  
\( A \) = magnitude of \( z \)  
\( j \) = imaginary unit  
\( \phi \) = phase in radians
<table>
<thead>
<tr>
<th>Property</th>
<th>Z-Transform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linearity</td>
<td>( a f_1[n] + b f_2[n] = a F_1(z) + b F_2(z) )</td>
</tr>
<tr>
<td>Shift left by ( k )</td>
<td>( f[n + k] = z^k F(z) - \sum_{n=0}^{k-1} f[n] z^{-n} )</td>
</tr>
<tr>
<td>Shift right by ( k )</td>
<td>( f[n - k] = z^{-k} F(z) )</td>
</tr>
<tr>
<td>Convolution</td>
<td>( f_1[n] f_2[n] = F_1(z) F_2(z) )</td>
</tr>
<tr>
<td>Final Value Theorem</td>
<td>( \lim_{n \to \infty} f[n] = \lim_{z \to 1} (z - 1) F(z) )</td>
</tr>
</tbody>
</table>
Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

C2000™ is a trademark of Texas Instruments. All trademarks are the property of their respective owners.
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
Copyright © 2022, Texas Instruments Incorporated