FTF 2016
TECHNOLOGY FORUM

MAKE DIGITAL POWER EASY USING KINETIS V SERIES POWER CONVERSION LIBRARY

FTF-SMI-N1949

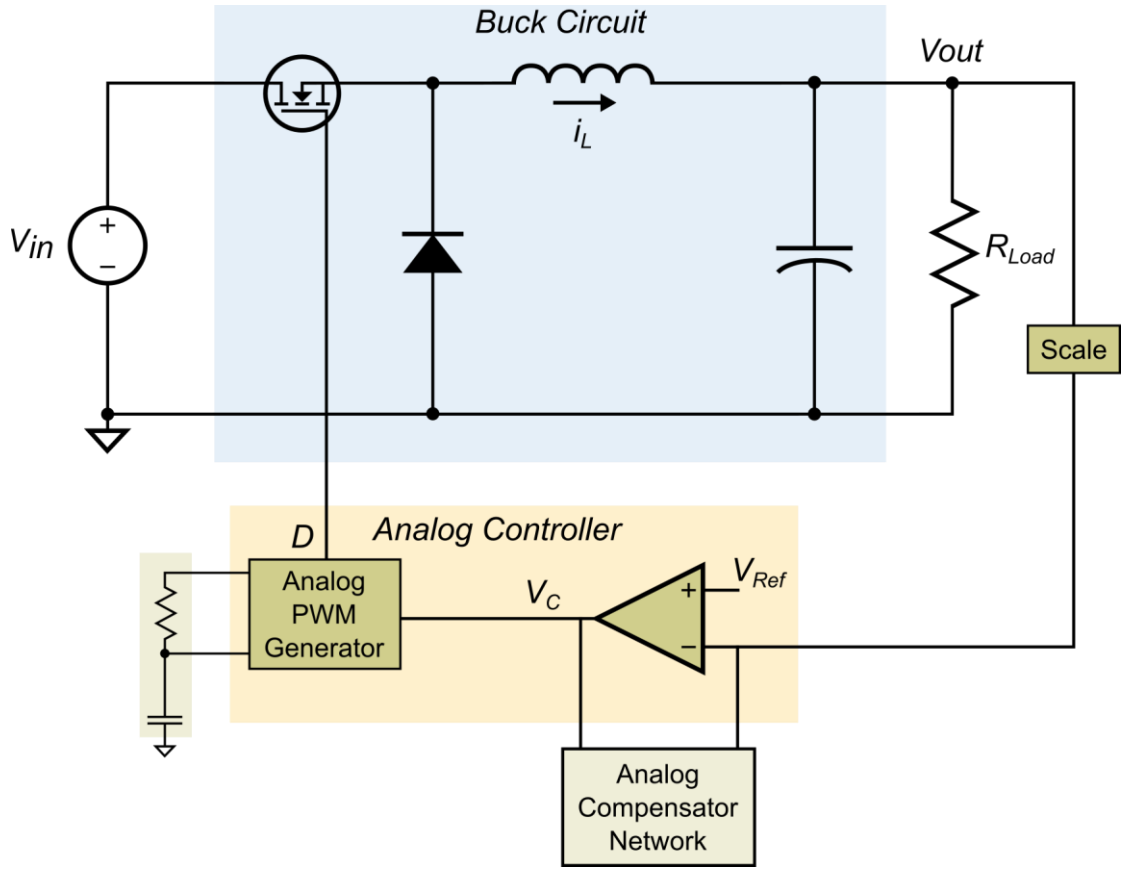
PAVEL GRASBLUM, PH.D.
TECHNICAL LEAD FOR SMART ENERGY SW GROUP
FTF-SMI-N1949
MAY 18, 2016

AGENDA

- Introduction to Digital Control of SMPS
- Brush up on the Analog Control Loop Design
- Digital Control Loop Design
- SW & HW Enablement for SMPS Design

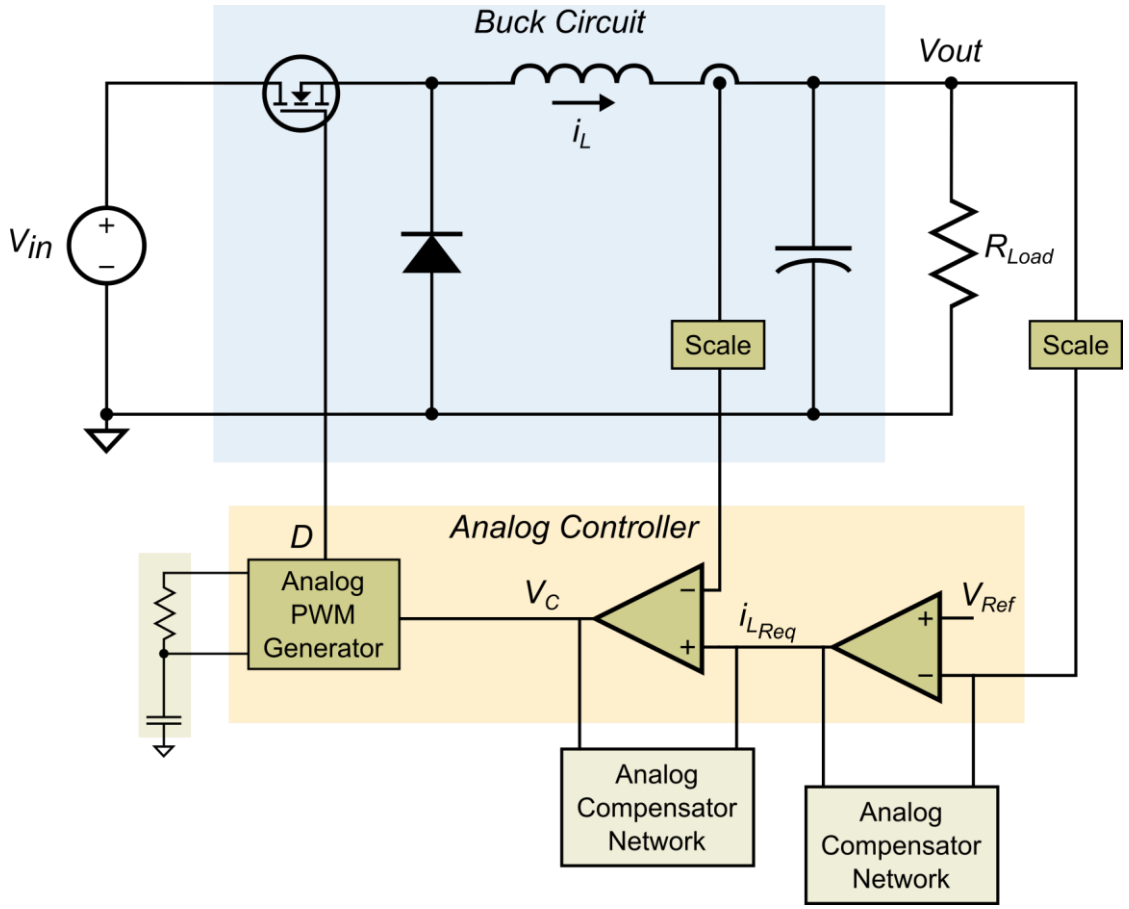
Analog Control

- SMPS circuit controlled by dedicated analog IC
- All parameters (PWM frequency, start up, fault thresholds, etc.) defined by passive components
- Circuit performance (compensator behavior) defined by passive components



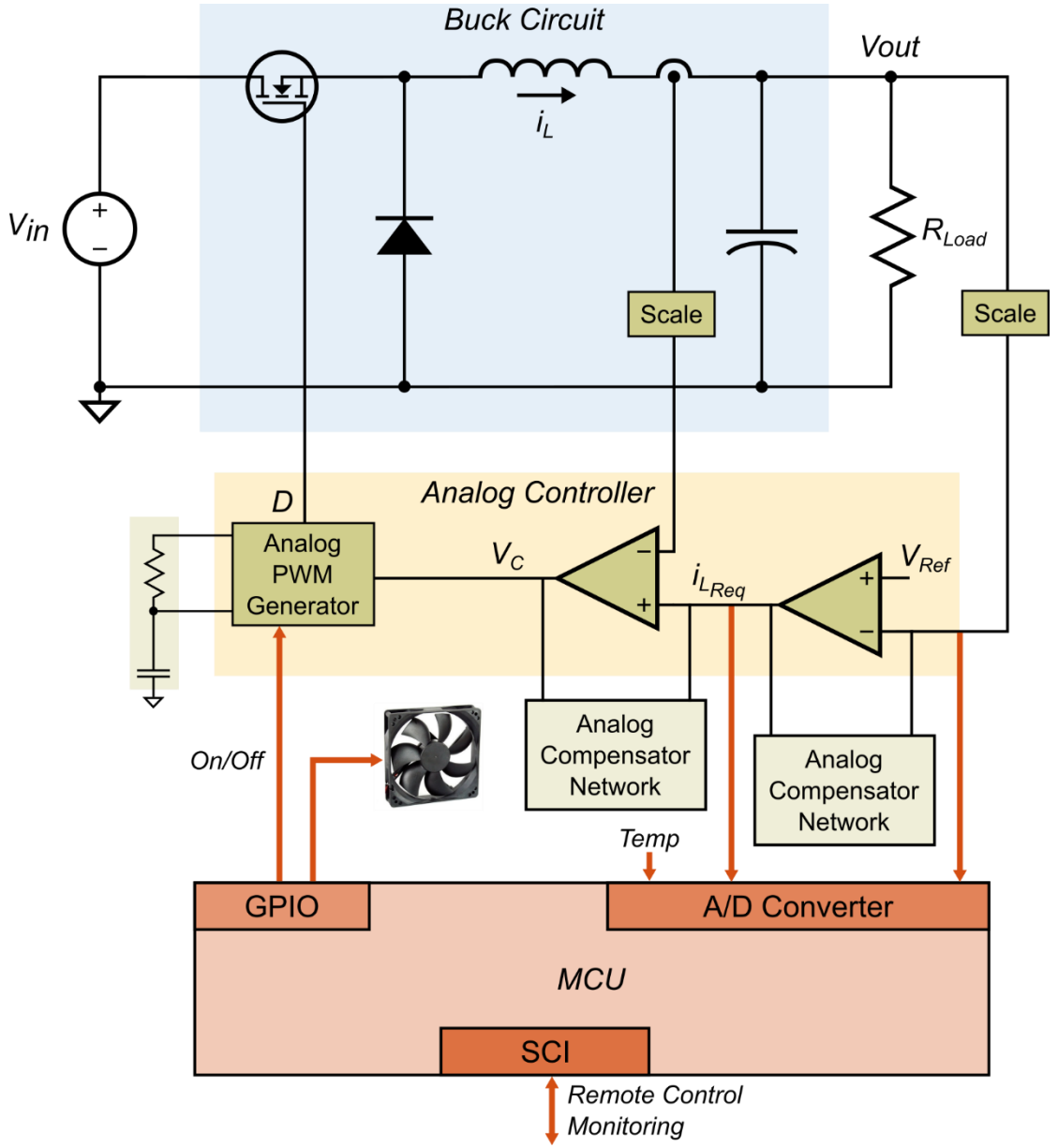
Analog Control

- SMPS circuit controlled by dedicated analog IC
- All parameters (PWM frequency, start up, fault thresholds, etc.) defined by passive components
- Circuit performance (compensator behavior) defined by passive components



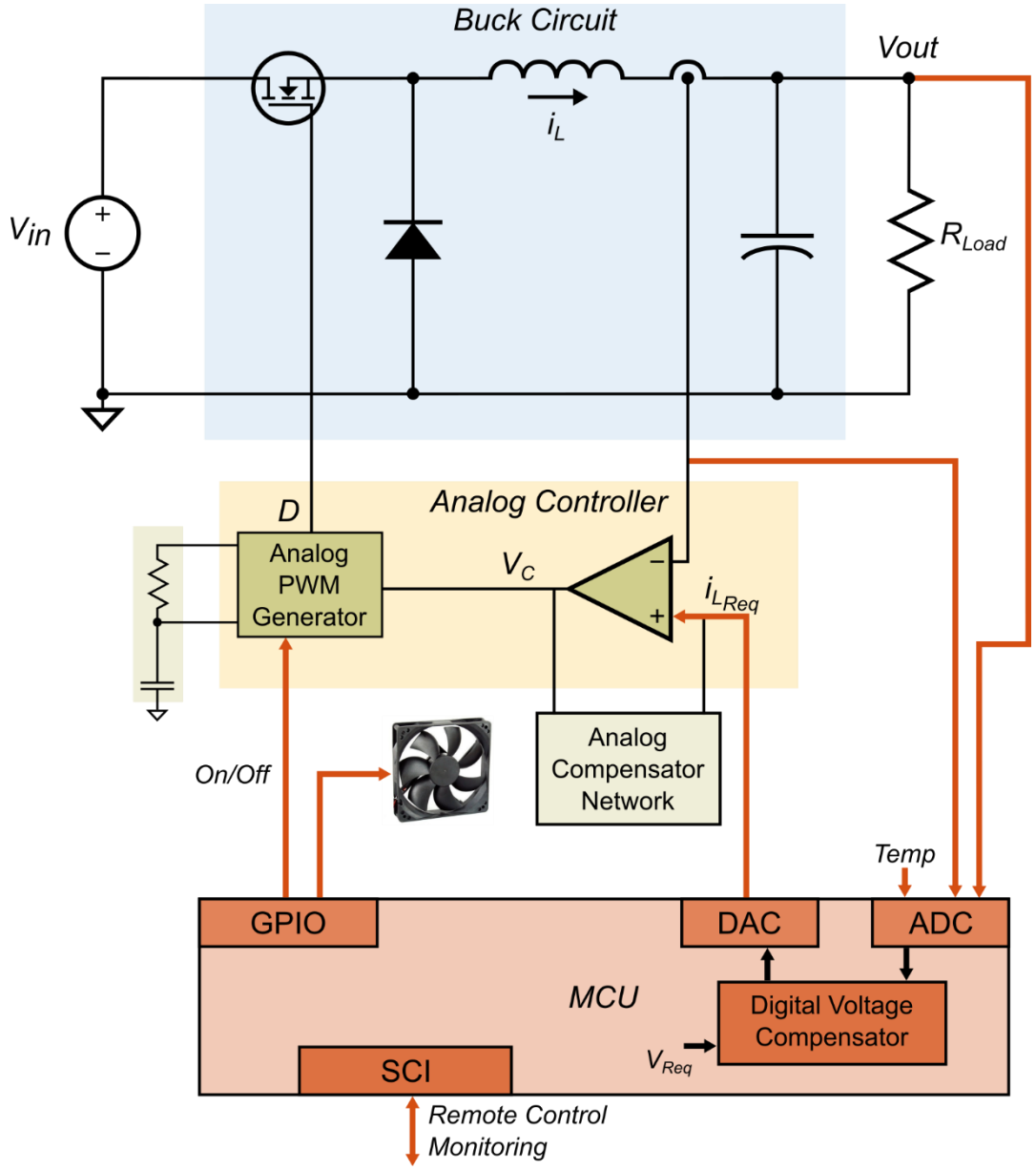
Analog Control with Digital Management

- SMPS controlled by dedicated analog IC
- MCU handles:
  - monitoring
  - remote control
  - cooling
  - diagnostic
- General purpose MCU
- MCU operation mostly independent to SMPS operation



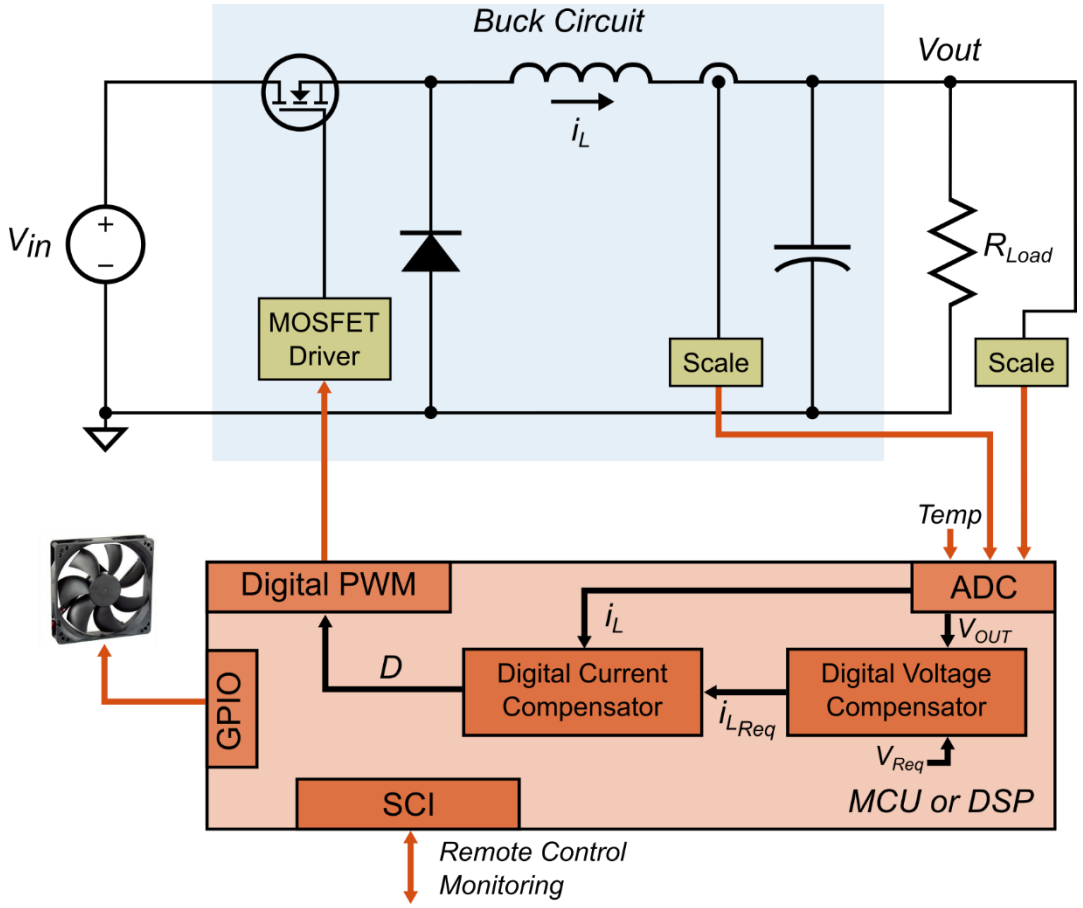
Analog/Digital Mixed Control

- Fast control loop controlled by analog circuit
- Slow control loop controlled by MCU
- General purpose MCU can be used
- No dedicated peripherals are needed
- Slow control loop fully digital



Fully Digital Control

- All control loops fully digital
- All signals processed or generated by MCU
- Allows to use non-linear or adaptive control algorithms
- Requires powerful MCU and dedicated peripherals



Benefits of Digital Power

- **Free from** the effects of **component tolerance**, parametric drift, aging, etc.
- **Configurable feedback loop structure** for specific application requirements
- **Adaptive control** to meet changing operating conditions
- **Programmable relationships** among **PWM** outputs
- **Upgradeable** with new features **without hardware** changes
- **Retainable** operational **data** for diagnostic and record keeping
- **Diverse communications** capabilities
- **Reduced** component count and **cost**
- **Higher** power density due to over all **integration**
- **Shorter R&D cycle**, fewer turns of board prototyping
- **Defendable firmware**–protects IP and differentiating technology

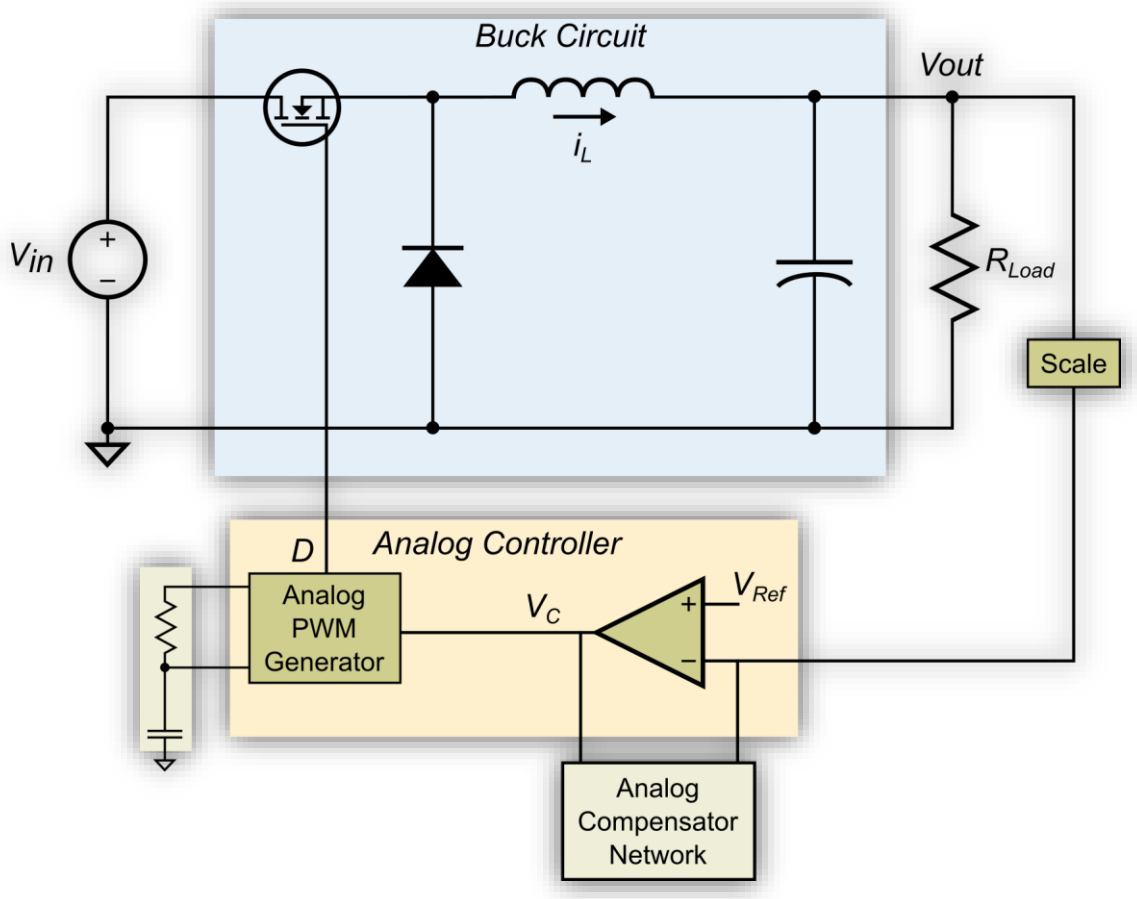
Limits of Digital Power

- Limited control bandwidth
  - Sampled control loop
  - Peripherals limitation (ADC, PWM module)
  - CPU performance
- Low integration
  - Bias power supplies, MOSFET drivers, signal conditioning
  - Higher price for small and simple SMPS
- Requires discrete control theory knowledge

BRUSH UP ON THE ANALOG CONTROL LOOP DESIGN

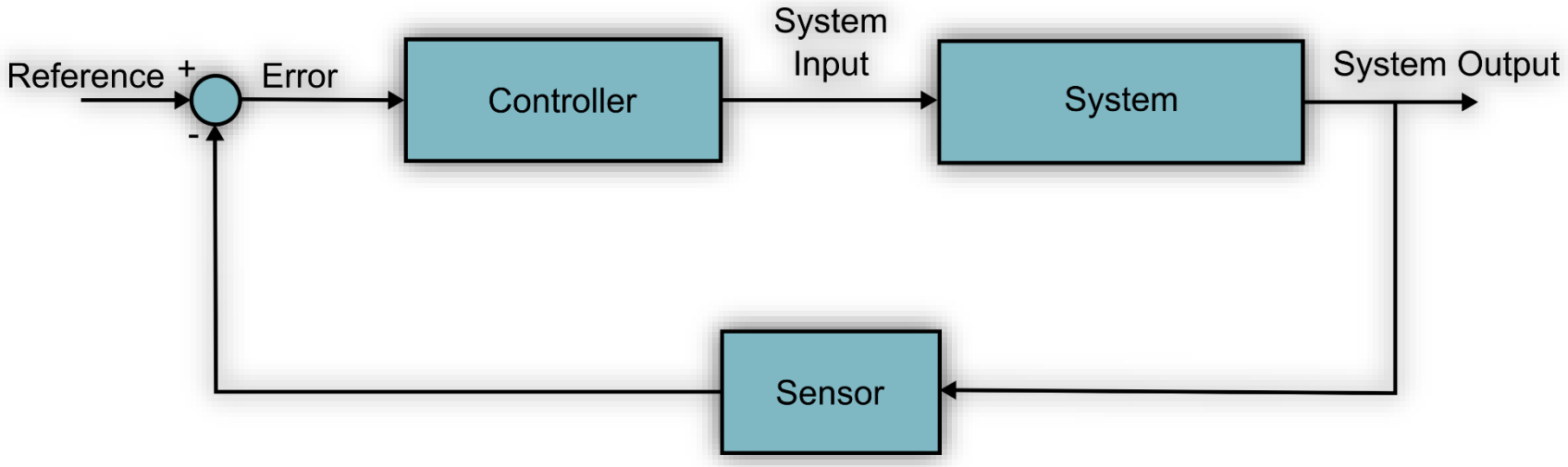
Analog Control Loop Design – Buck Converter

- Example of analog SMPS control
  - Buck converter
  - Voltage mode control
  - We will use Type II controller to design stable control loop



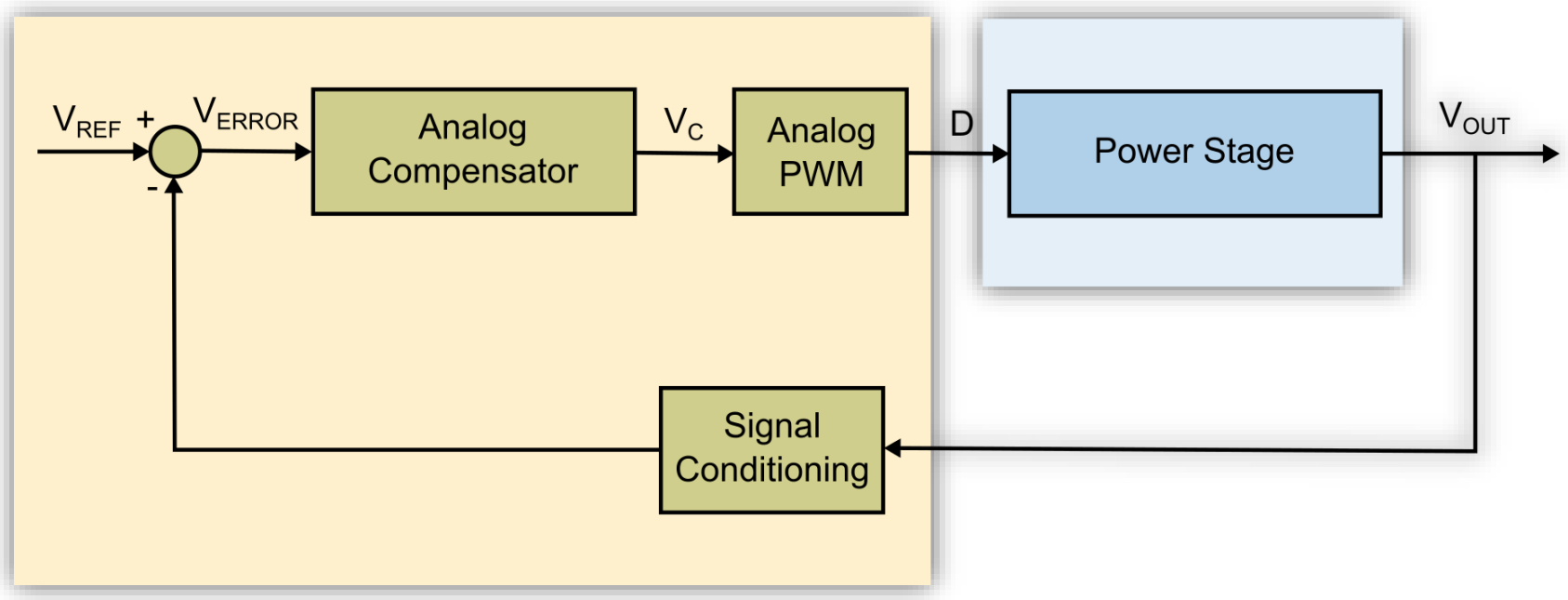
Analog Control Loop Design – Buck Converter

- Closed-loop control system



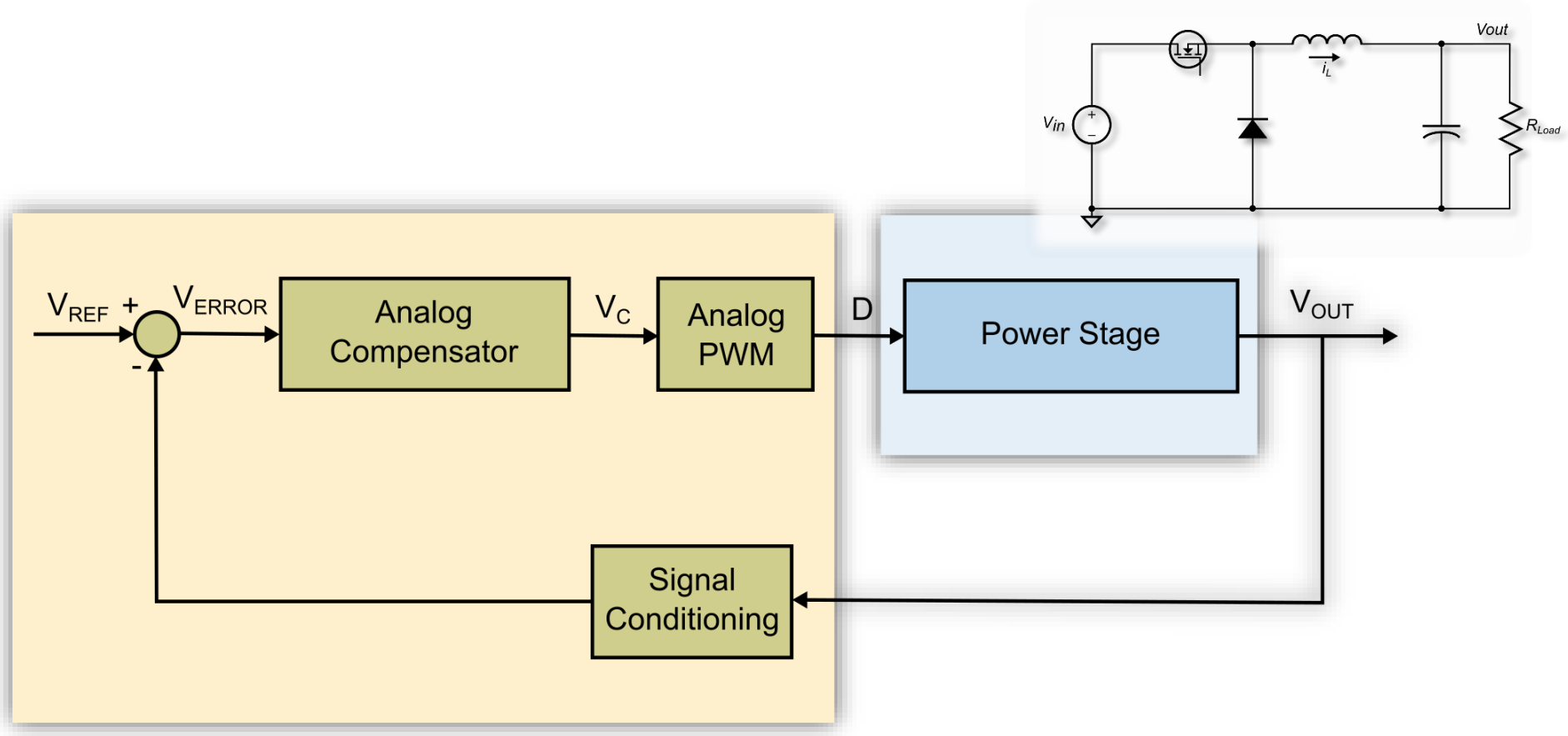
Analog Control Loop Design – Buck Converter

- Buck converter closed-control loop



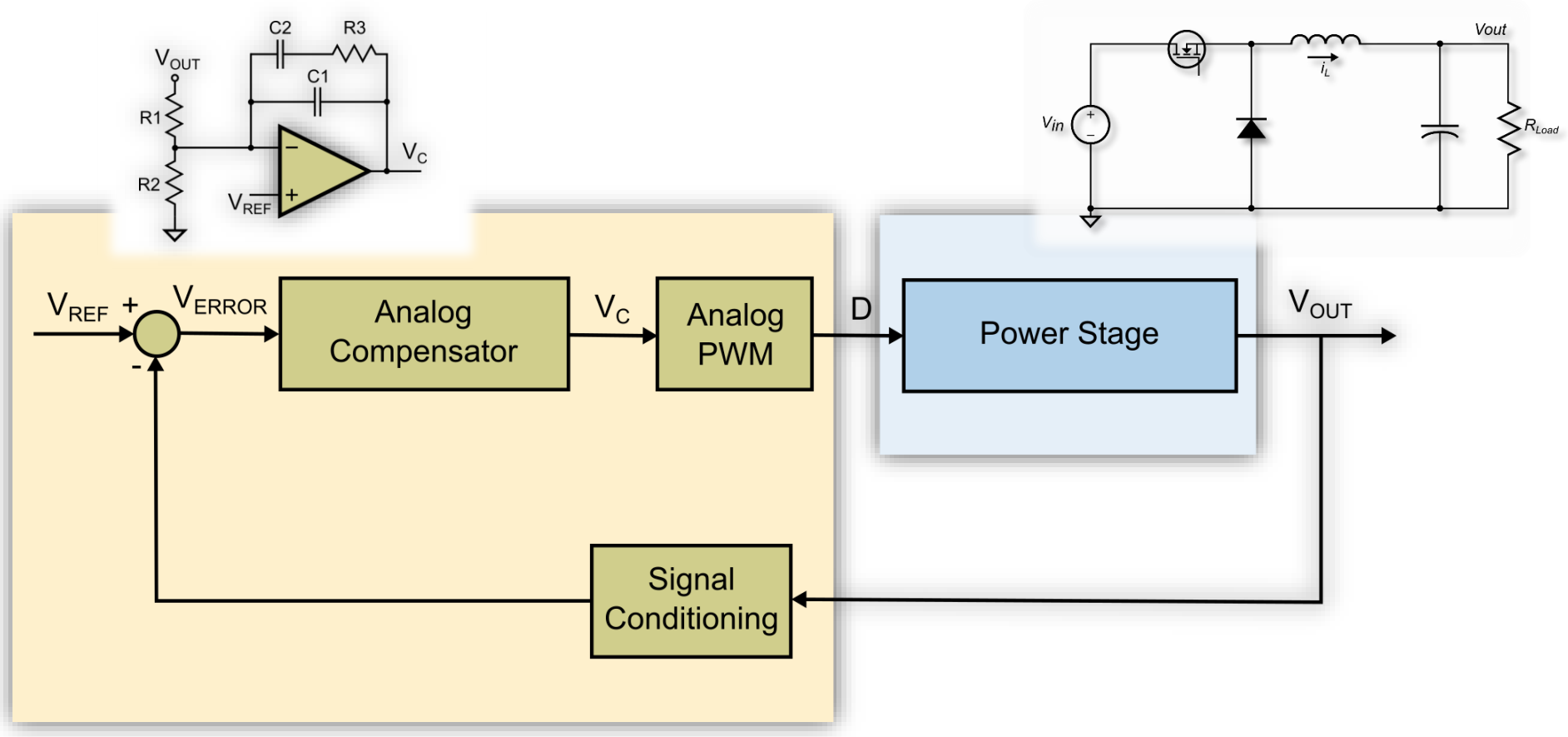
Analog Control Loop Design – Buck Converter

- Buck converter closed-control loop



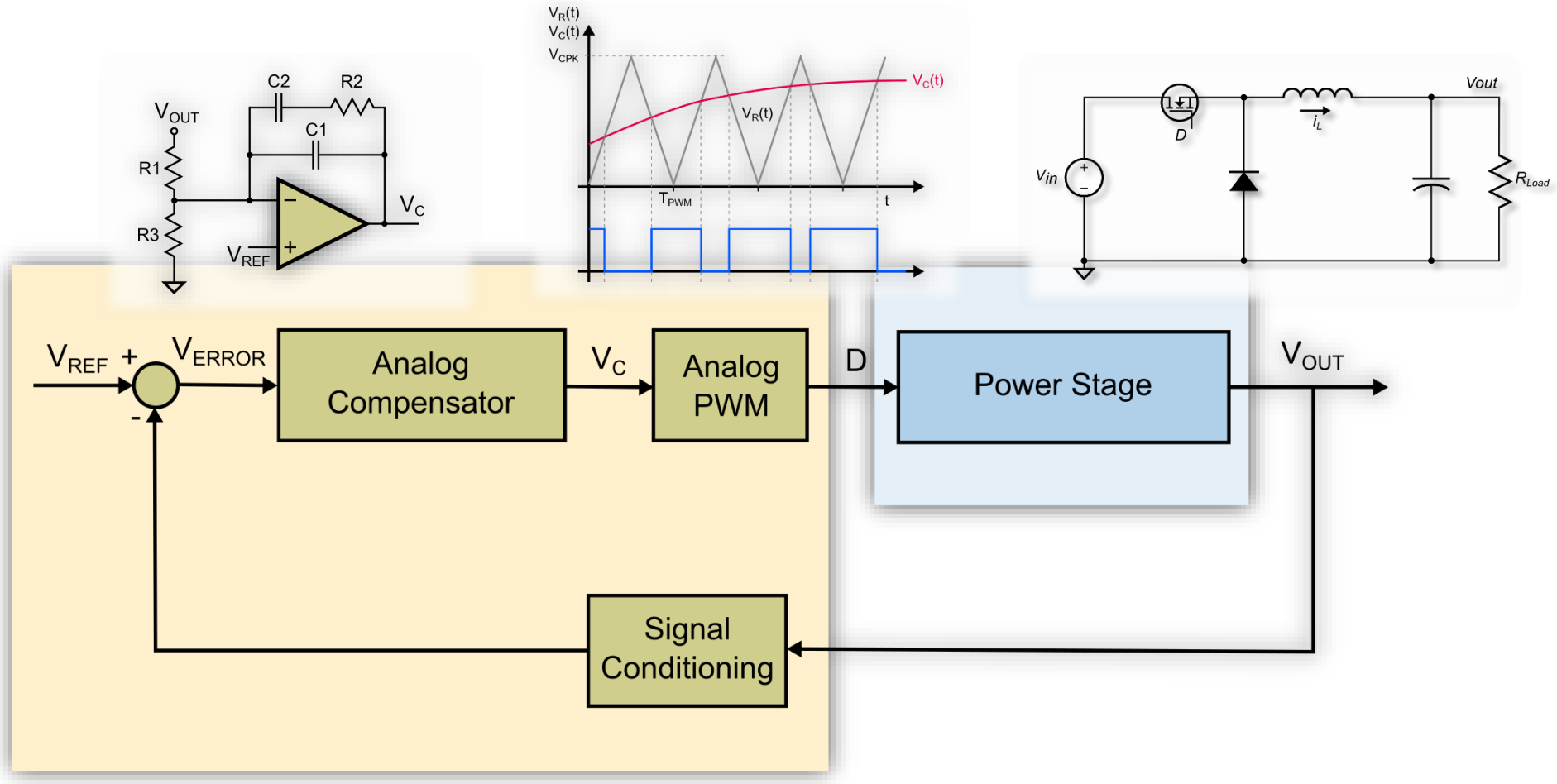
Analog Control Loop Design – Buck Converter

- Buck converter closed-control loop



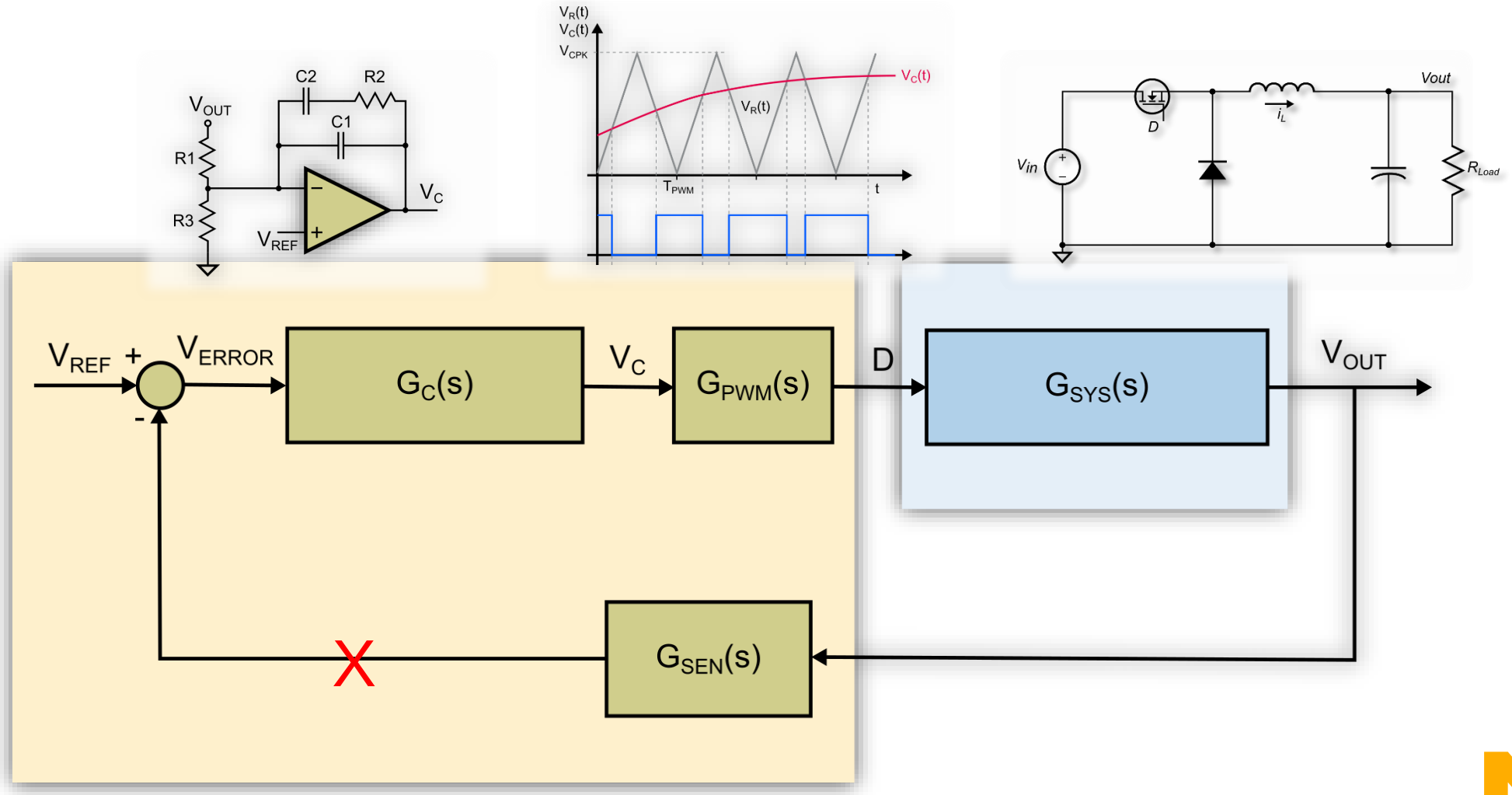
Analog Control Loop Design – Buck Converter

- Buck converter closed-control loop



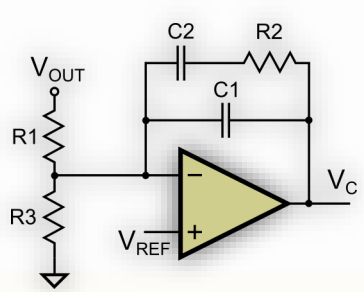
Analog Control Loop Design - Buck Converter Example

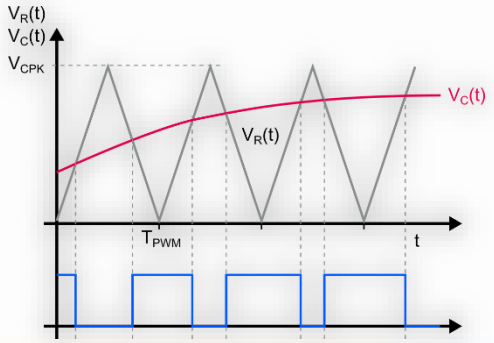
- Buck converter open-control loop

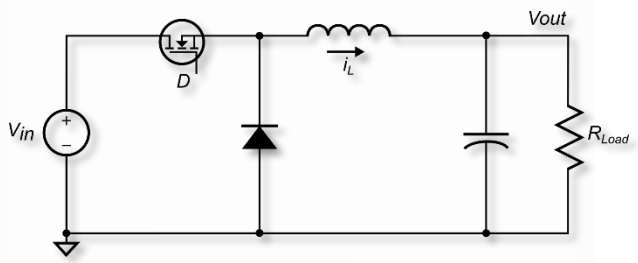


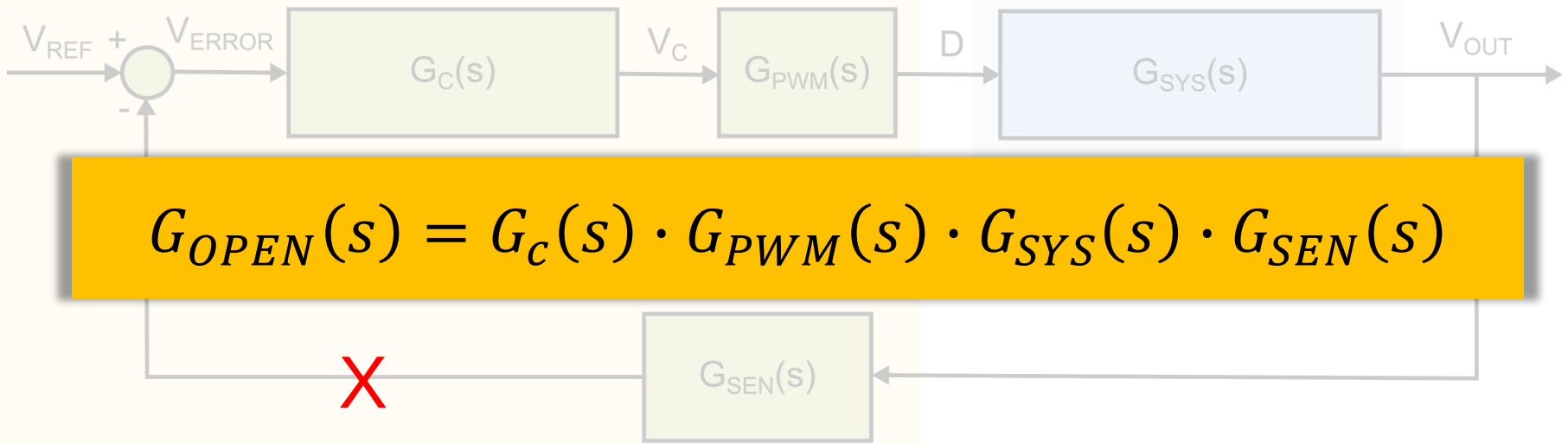
Analog Control Loop Design - Buck Converter Example

- Buck converter open-control loop









$$G_{OPEN}(s) = G_c(s) \cdot G_{PWM}(s) \cdot G_{SYS}(s) \cdot G_{SEN}(s)$$

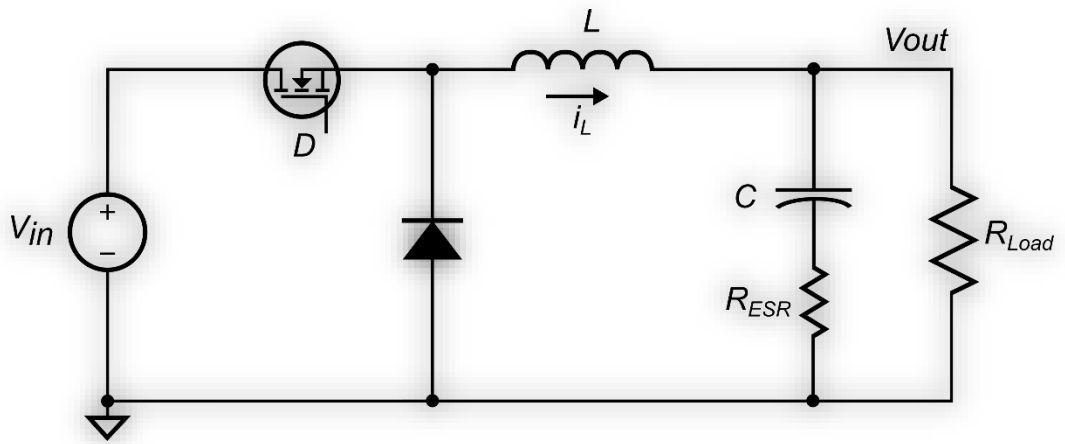
Analog Control Loop Design - Buck Converter Example

- How to get transfer function of buck converter?

Averaging over switching period

Perturbation and Linearization

Small signal AC model

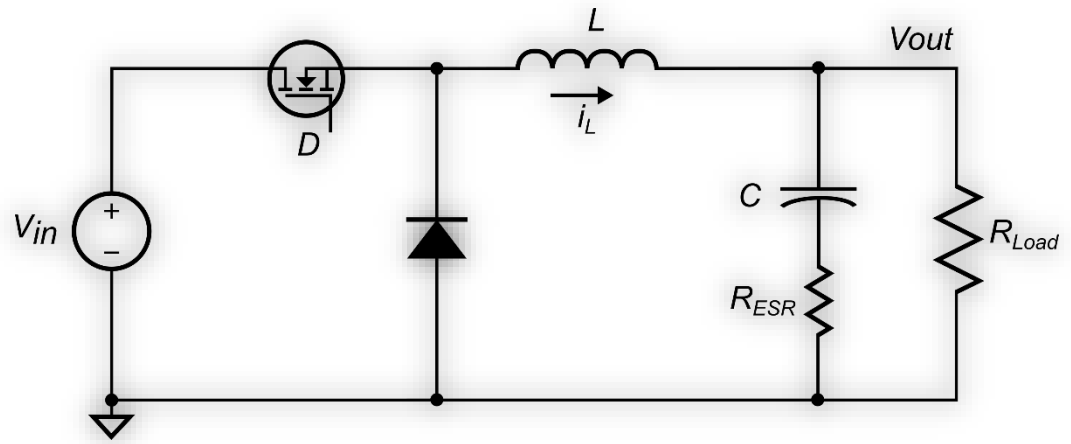


$$G_{SYS}(s) = \frac{v_{out}(s)}{d(s)} = \frac{R_{LOAD} v_{in}(1 + sR_{ESR}C)}{s^2 LC(R_{LOAD} + R_{ESR}) + s(L + R_{LOAD}R_{ESR}C) + R_{LOAD}}$$

Analog Control Loop Design - Buck Converter Example

- Circuit parameters

$v_{in} = 12\ \mathrm{V}$
$v_{out} = 5\ \mathrm{V}$
$R_{load} = 0.5\ \Omega$
$L = 4.8\ \mu\mathrm{H}$
$C = 2040\ \mu\mathrm{F}$
$R_{ESR} = 0.02\ \Omega$
$f_{sw} = 200\ \mathrm{kH}$



$$G_{SYS}(s) = \frac{v_{out}(s)}{d(s)} = \frac{47483.3808(s + 2.451 \cdot 10^4)}{s^2 + 4900s + 9.698 \cdot 10^7}$$

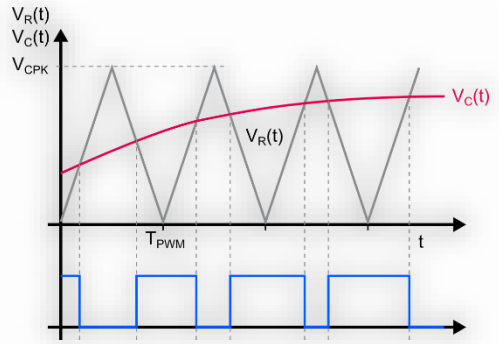
Analog Control Loop Design - Buck Converter Example

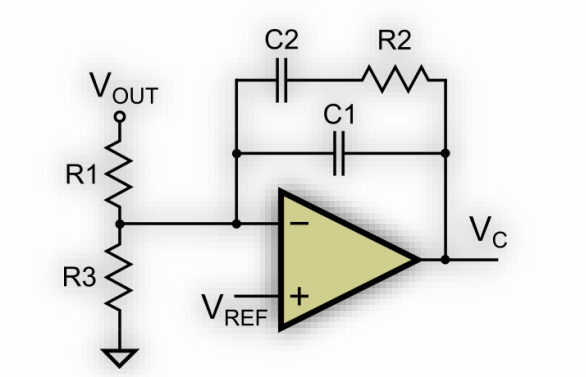
- Buck converter closed-control loop

- $G_{PWM}(s) = \frac{1}{V_{CPK}}$

- $G_{SEN}(s) = \frac{R_3}{R_1+R_3}$

- $G_C(s) = \frac{1}{R_1 C_1} \frac{s+\frac{1}{R_2 C_2}}{s(s+\frac{C_1+C_2}{R_2 C_1 C_2})}$





Type II Controller

Analog Control Loop Design - Buck Converter Example

- Open loop transfer function

  Lets $G_{PWM}(s) = 1$ and $G_{SEN}(s) = 0.2$ then

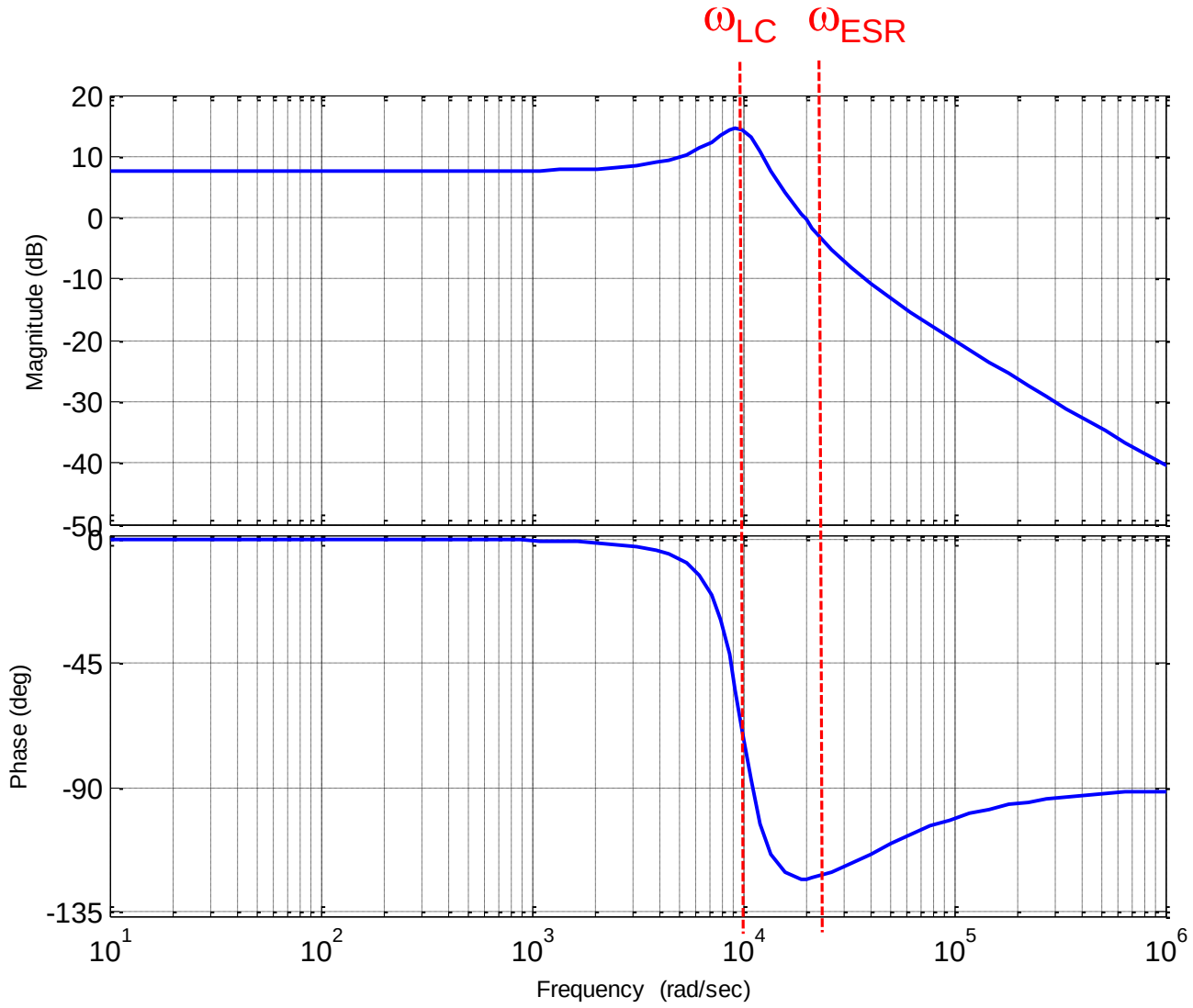
$$G_{OPEN}(s) = \frac{v_{out}(s)}{v_{ref}(s)} = \underbrace{\frac{K\left(1 - \frac{s}{z_1}\right)}{s\left(1 - \frac{s}{p_2}\right)}}_{G_C(s)} \underbrace{1}_{G_{PWM}(s)} \underbrace{\frac{1}{5}}_{G_{SEN}(s)} \underbrace{\frac{47483.3808\left(s + 2.451 \cdot 10^4\right)}{s^2 + 4900s + 9.698 \cdot 10^7}}_{G_{SYS}(s)}$$

Analog Control Loop Design - Buck Converter Example

- Controller Selection - $G_C(s)$
  - Performance parameters of closed loop
    $\omega_c = 0.05 f_{sw} = 62.831 \cdot 10^3 \text{ rad/s}$
    $PM > 45°$
  - Controller Type II selected

$$G_C = \frac{K(1 - \frac{s}{z_1})}{s\,(1 - \frac{s}{p_2})}$$

  - Place z_1 at $0.75\omega_{LC} = 7532 \text{ rad/s}$
  - Place p_2 at $\omega_2 = \frac{1}{2} 2\pi f_{sw} = 6.28 \cdot 10^5 \text{ rad/s}$
  - Adjust K to get desired loop bandwidth at ω_C

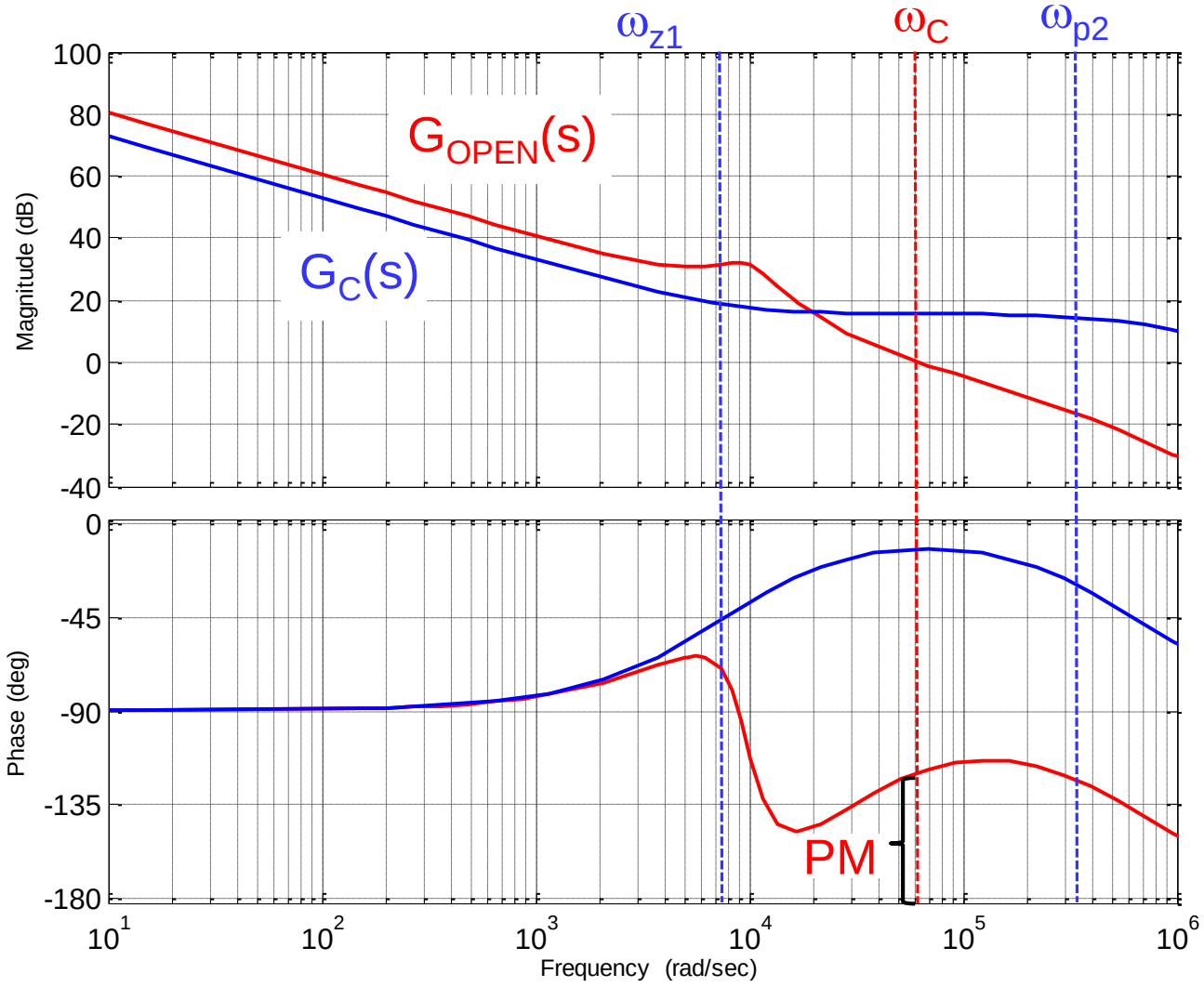


Analog Control Loop Design - Buck Converter Example

- Complete open loop $G_{OPEN}(s)$

$$G_{PWM}(s) \cdot G_{SYS}(s) \cdot G_{SEN}(s) \cdot G_C(s)$$

- $G_C = \frac{3781584(s+7532)}{s\,(s+6.283 \cdot 10^5)}$
- $\omega_C = 62.831 \cdot 10^3$ rad/s
- $PM = 60.7°$



Analog Control Loop Design - Buck Converter Example

- Now we can calculate all passive components

R_1 choosen to be 4.7 kΩ

$$R_3 = \frac{R_1 V_{REF}}{(V_{out} - V_{REF})} = 1175\ \Omega$$

$$R_2 = \frac{R_1 V_{CPK} \omega_{ESR} \omega_C}{V_{in} \omega_{LC}^2} = 5906\ \Omega$$

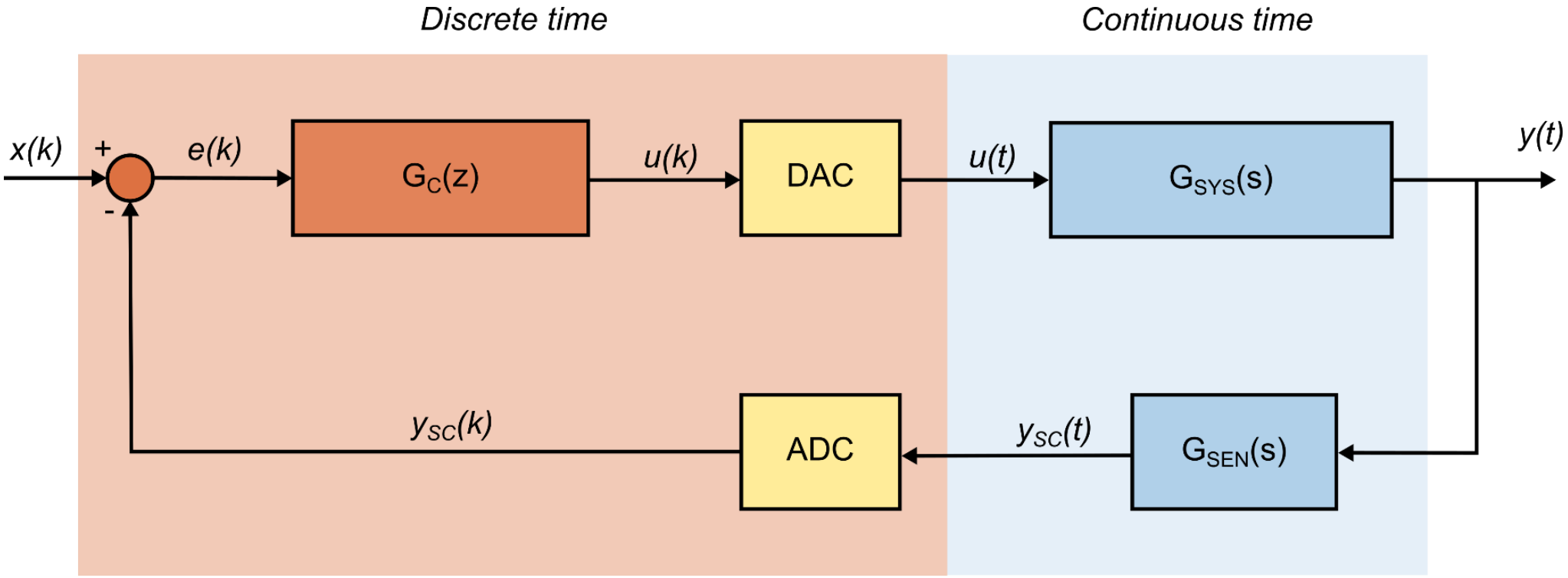
$$C_2 = \frac{1}{R_2 \omega_{z_1}} = 22.48\ \text{nF}$$

$$C_1 = \frac{1}{R_2 \omega_{p_2}} = 270\ \text{pF}$$

DIGITAL CONTROL LOOP DESIGN IN SWITCHED MODE POWER SUPPLIES

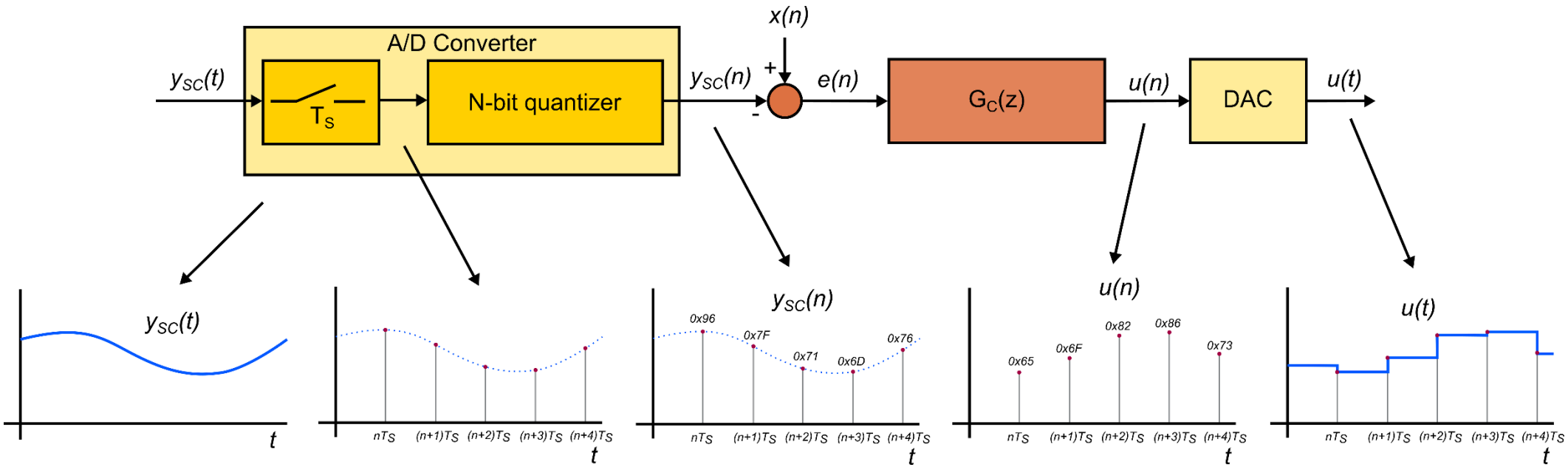
Digital Control Loop Design

- Digital Control Loop



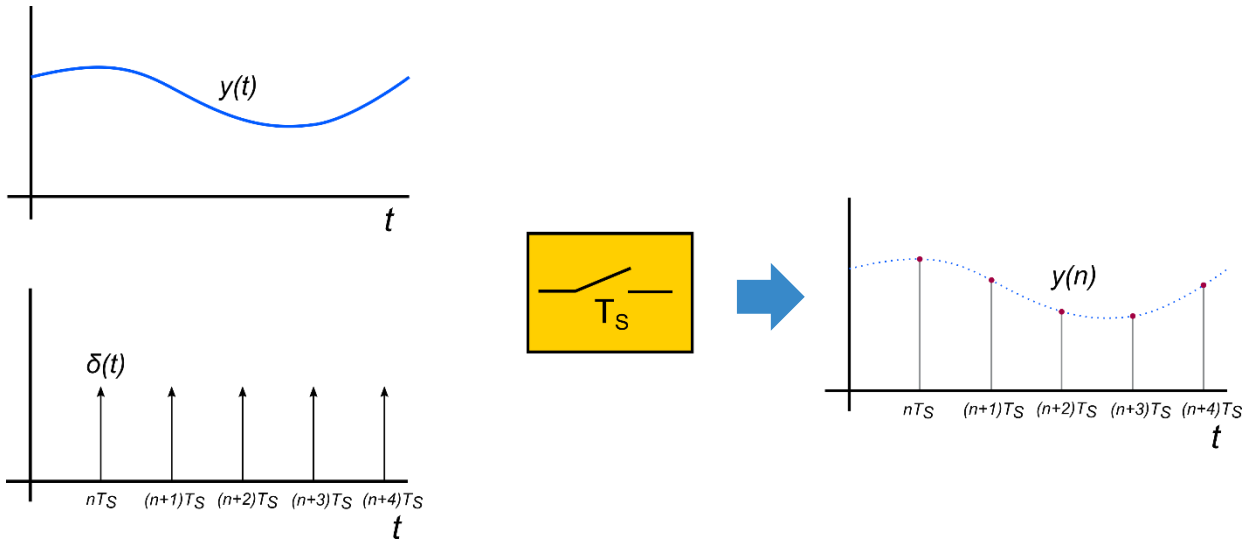
Digital Control Loop Design – Sampling

- Digital Control Loop



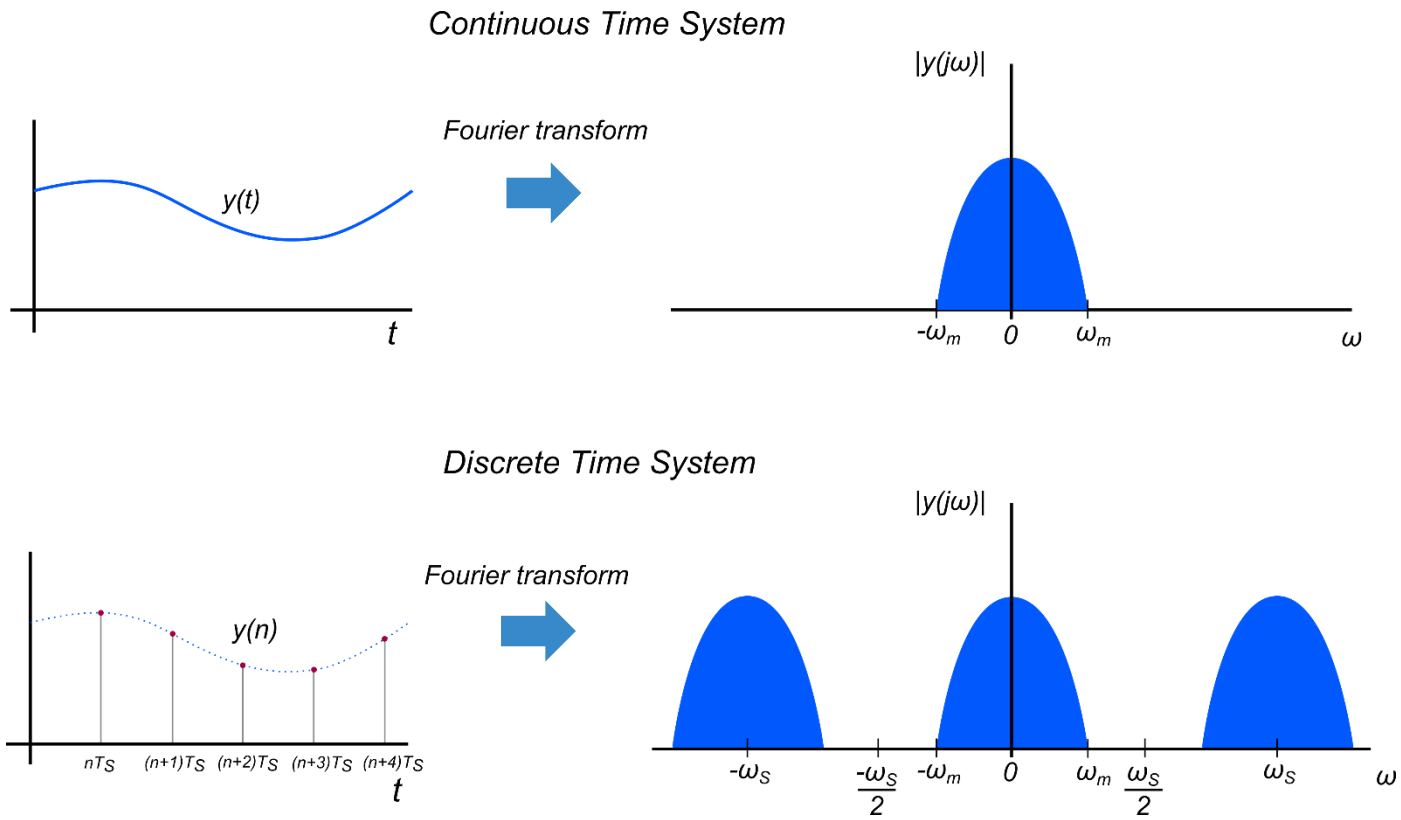
Digital Control Loop Design – Sampling

- Sampling converts continuous-time domain into discrete time domain



Digital Control Loop Design – Sampling

- Sampling replicates frequency spectrum of continuous system



Digital Control Loop Design – Sampling

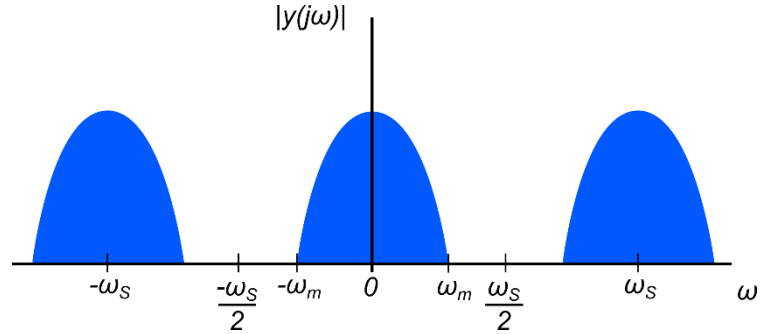
- Aliasing

  To be able to reconstruct samples back to continuous system the sampling frequency ω_S has to be at least twice larger than highest frequency of sampled system (ω_m)

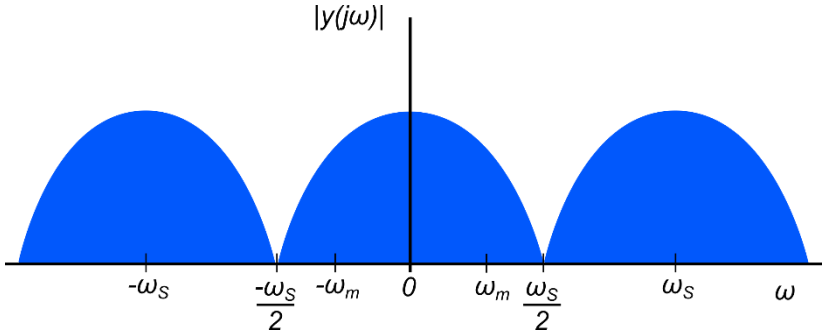
$$\omega_S > 2\omega_m$$

Nyquist frequency

$$\frac{\omega_S}{2} > \omega_m$$



$$\frac{\omega_S}{2} = \omega_m$$



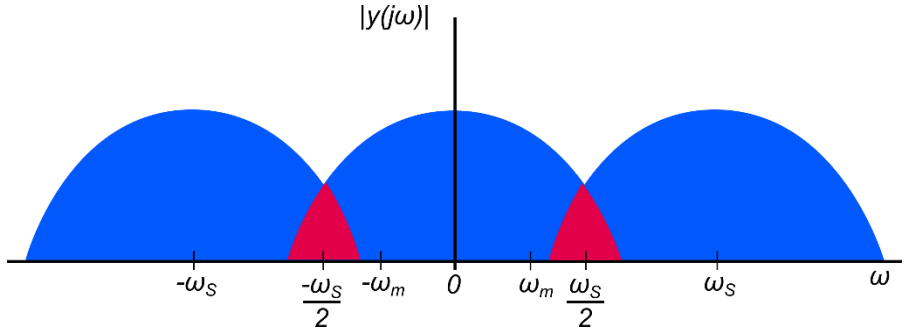
In both cases no aliasing observed

Digital Control Loop Design – Sampling

- Aliasing

  If $\omega_S < 2\omega_m$ the spectrums due to sampling overlaps and there is not possible to reconstruct samples back into continuous time domain

$$\frac{\omega_S}{2} < \omega_m$$

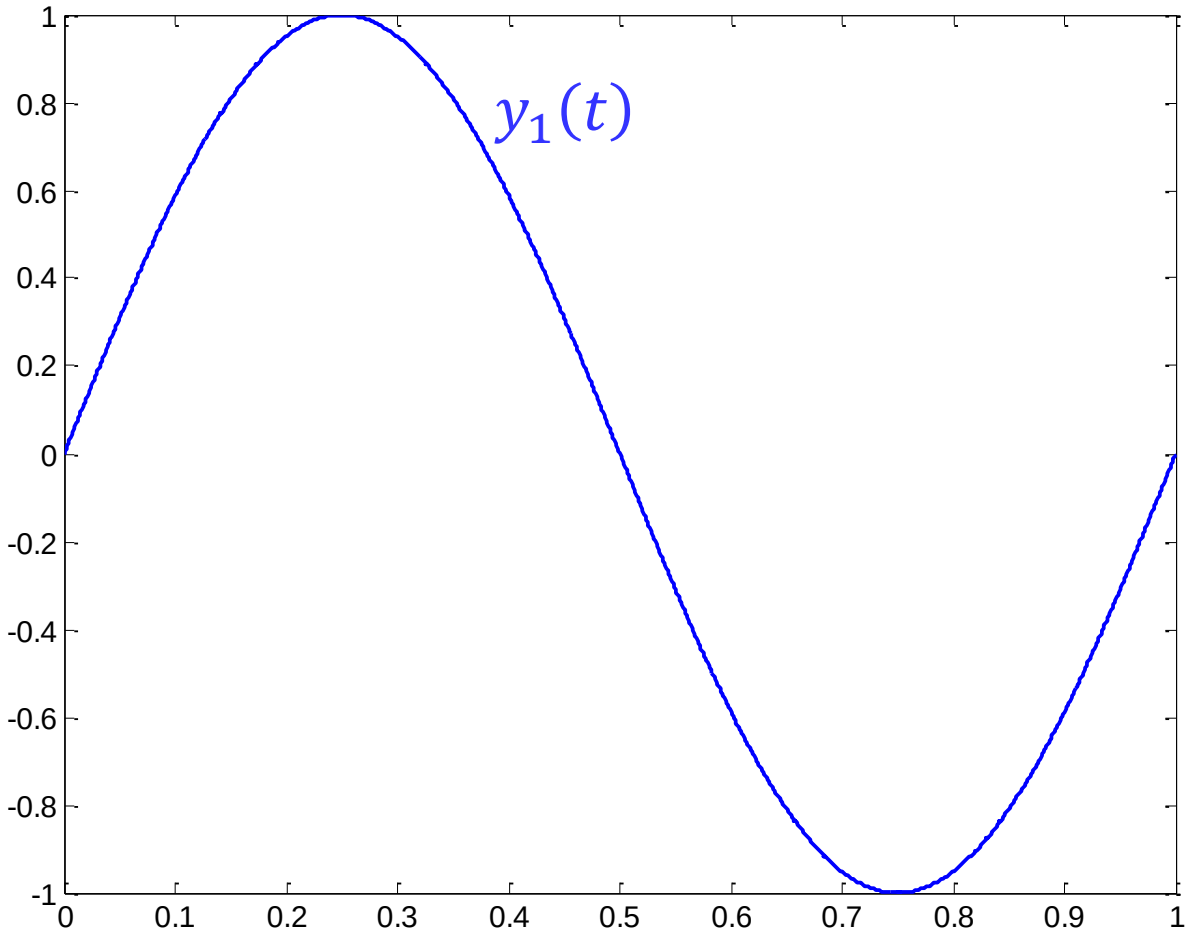


Aliasing is present

Digital Control Loop Design – Sampling

- Aliasing example
  - Let's have two signals:

  $y_1(t) = \sin 2\pi t$ (1Hz)

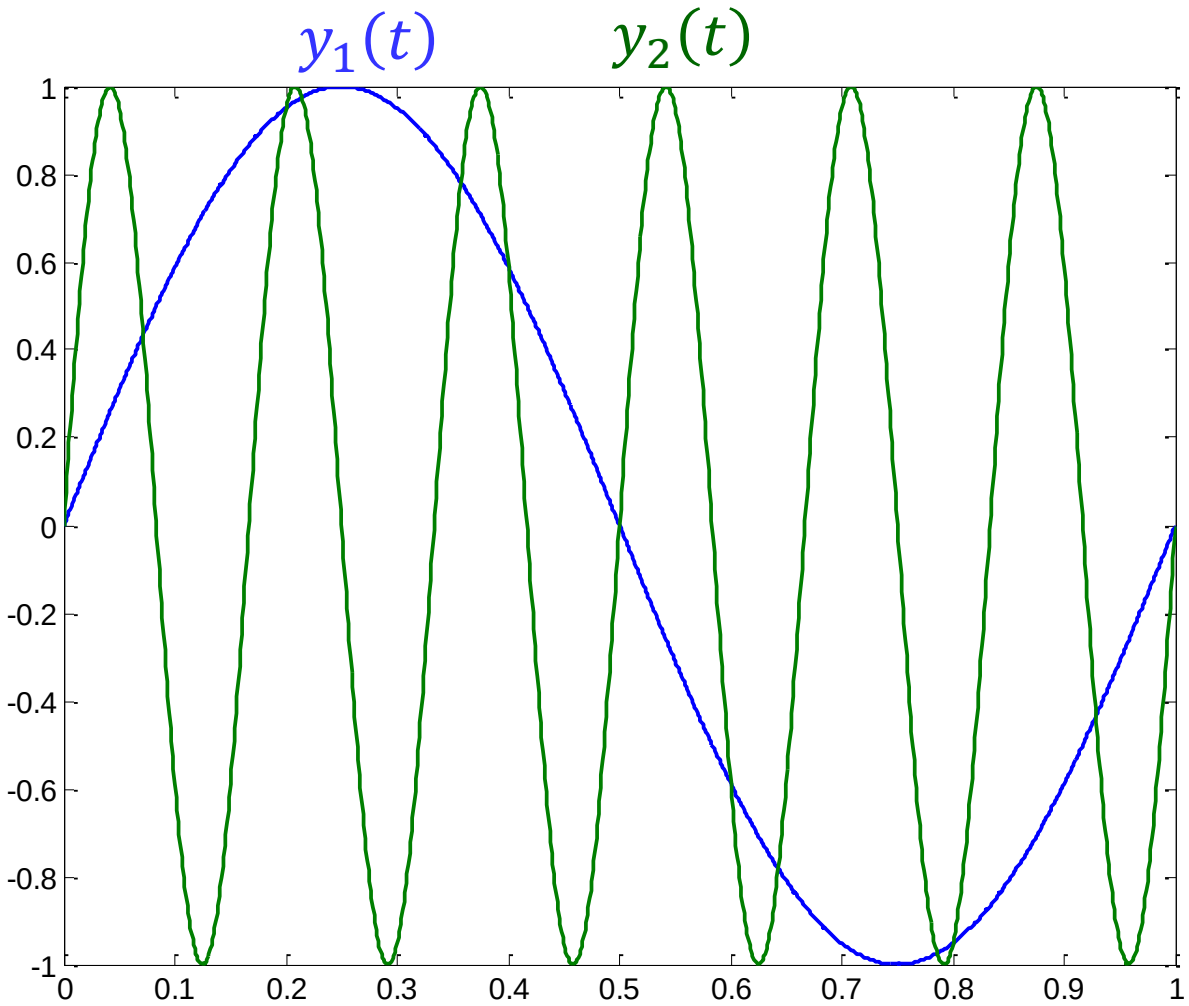


Digital Control Loop Design – Sampling

- Aliasing example
  - Let's have two signals:

  $y_1(t) = \sin 2\pi t$ (1Hz)

  $y_2(t) = \sin 12\pi t$ (6 Hz)

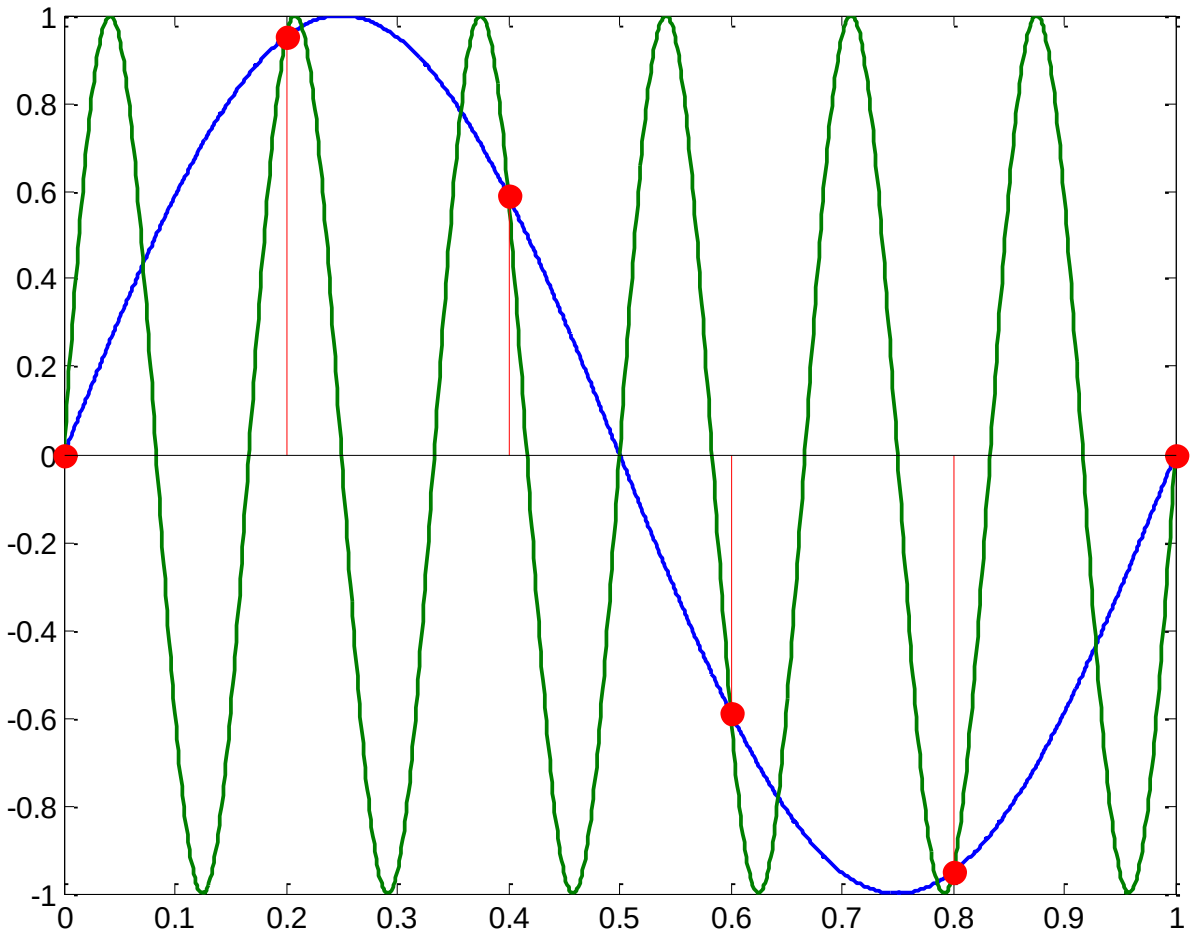


Digital Control Loop Design – Sampling

- Aliasing example
  - Let's have two signals:

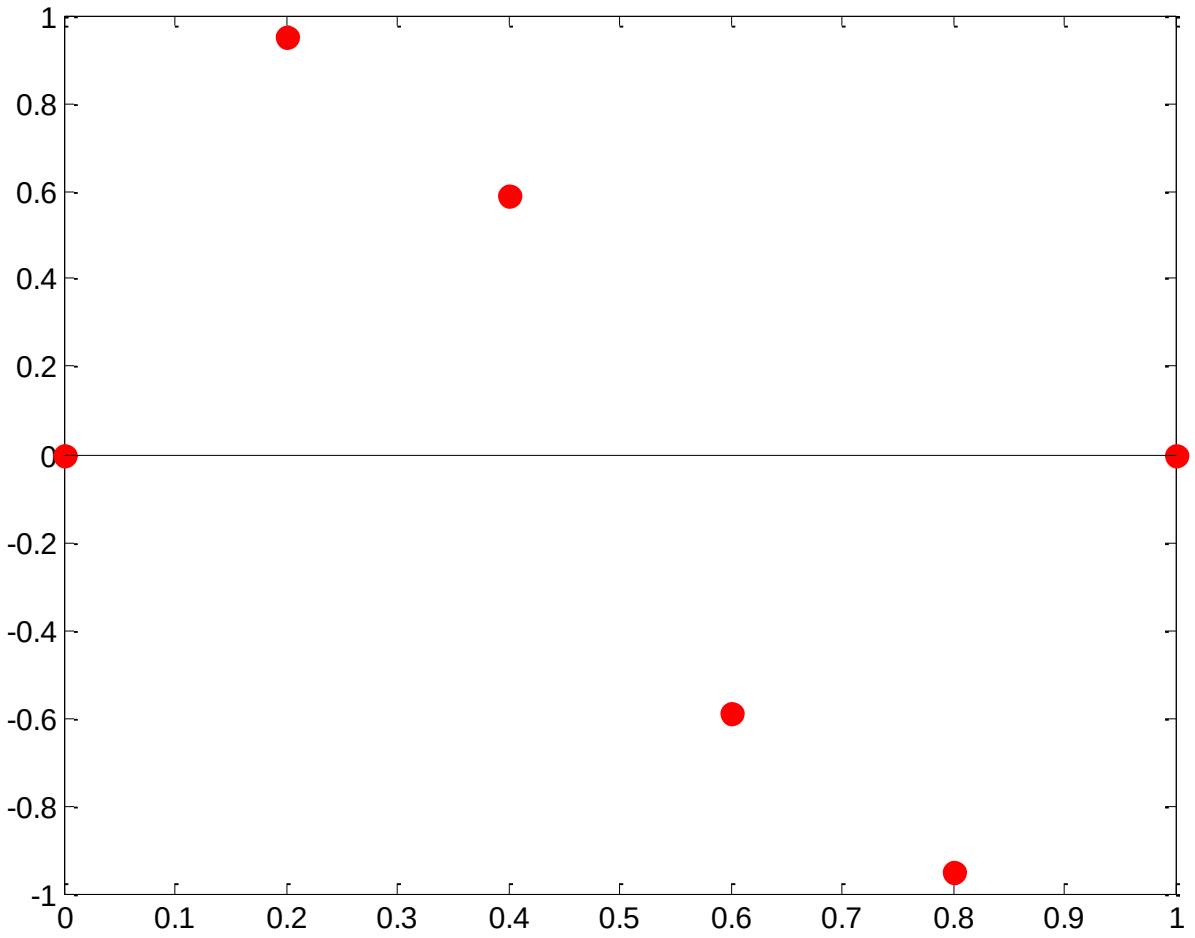
    $y_1(t) = \sin 2\pi t$ (1Hz)

    $y_2(t) = \sin 12\pi t$ (6 Hz)
  - Sample both signals with $\omega_S = 10\pi$ (5 Hz)



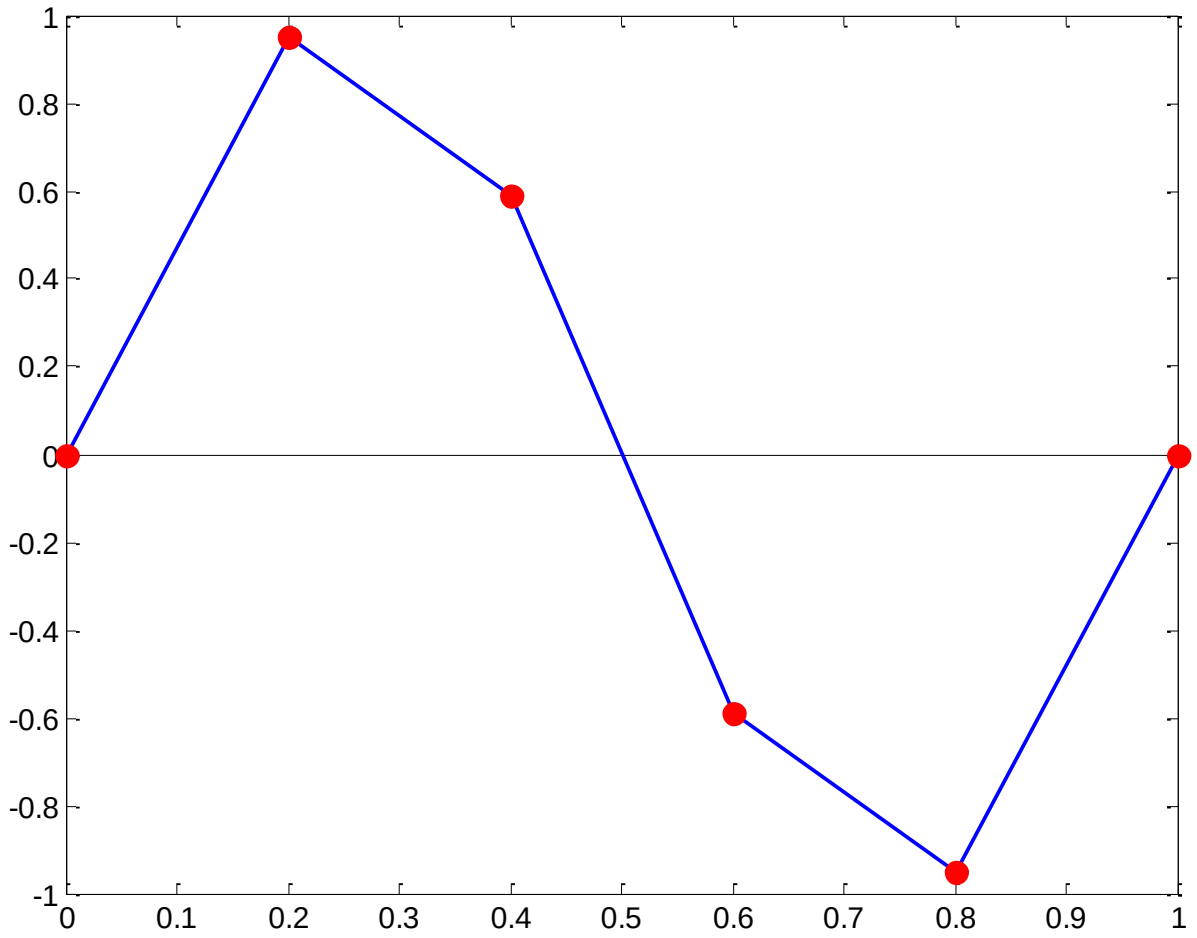
Digital Control Loop Design – Sampling

- Aliasing example
  - Reconstruct samples back into continuous time domain



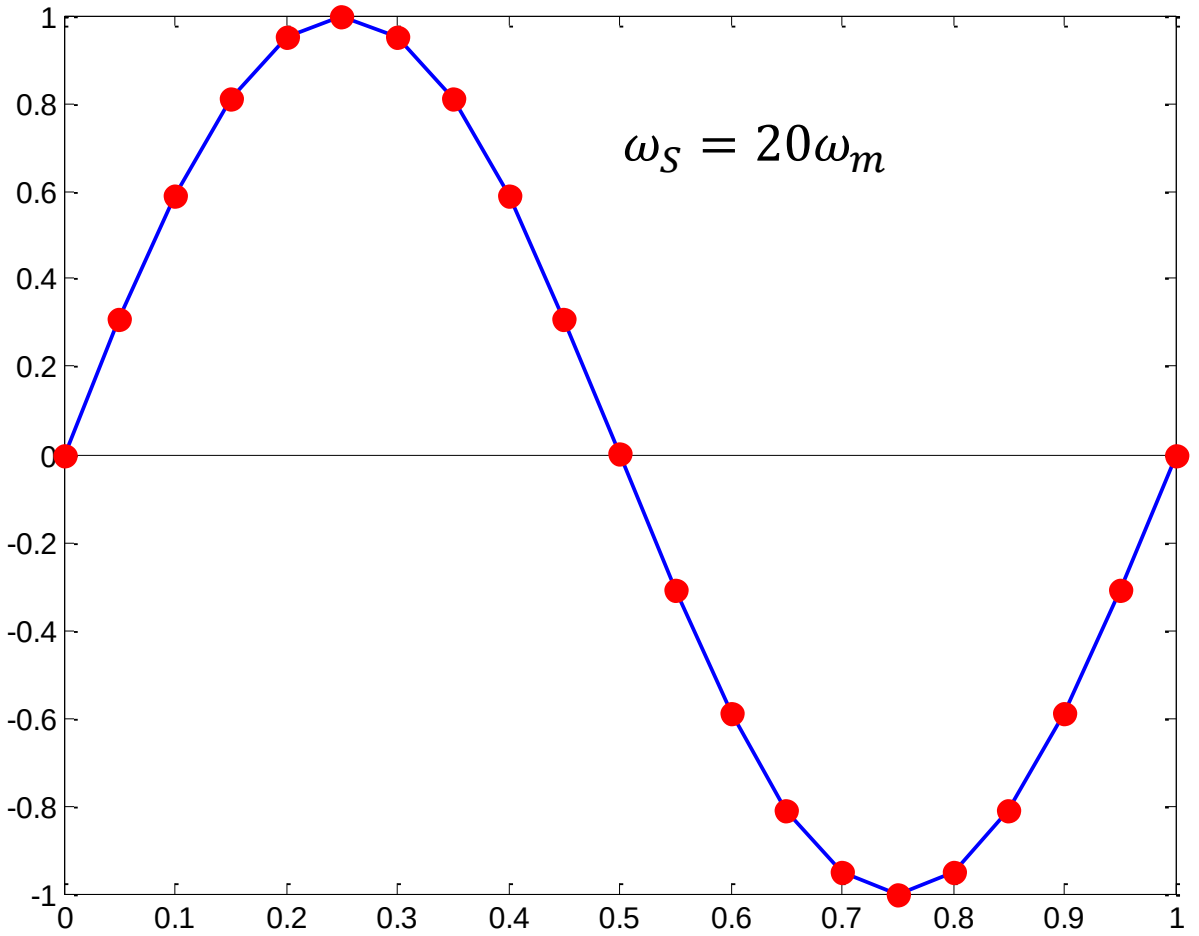
Digital Control Loop Design – Sampling

- Aliasing example
  - We are very close to y_1 (1 Hz), but y_2 (6 Hz) completely disappeared
  - Sampling 6 Hz signal with 5 Hz sampling frequency (below Nyquist frequency) leads to complete wrong reconstruction back into continuous system



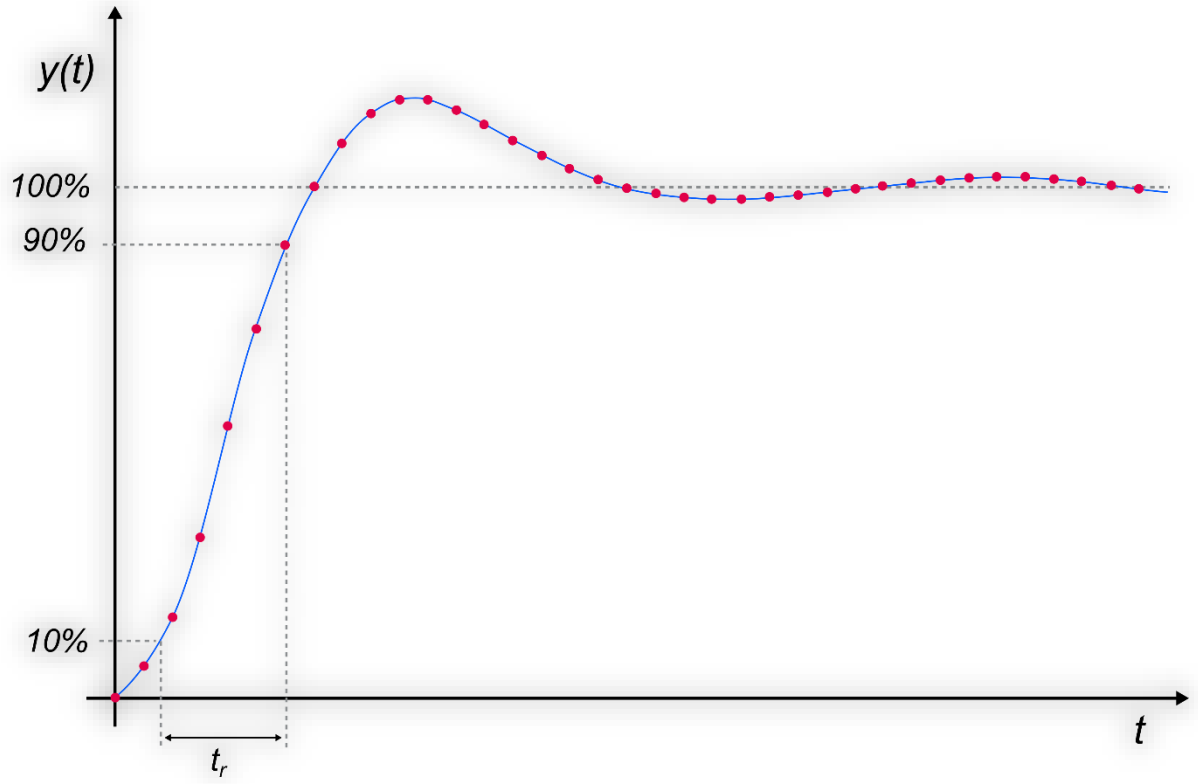
Digital Control Loop Design – Sampling

- How fast to sample?
  - Sampling with frequency $\omega_S > 2\omega_m$ is necessary but no sufficient condition for correct sampling



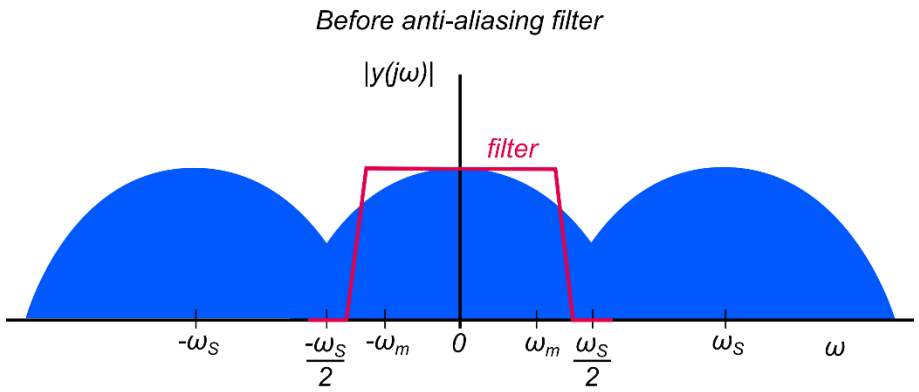
Digital Control Loop Design – Sampling

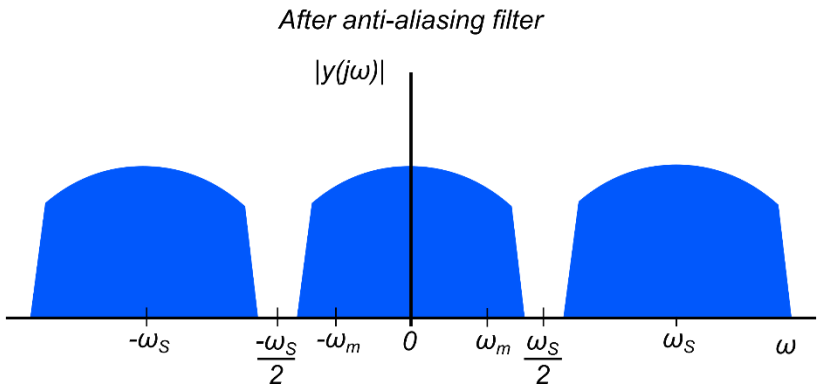
- How fast to sample?
  - Sampling with frequency $\omega_S > 2\omega_m$ is necessary but no sufficient condition for correct sampling
  - There should be at least 5-10 samples for highest frequency of the interest in the system
  - For control systems it corresponds at least 5-10 samples during rise time t_r of the system response



Digital Control Loop Design – Sampling

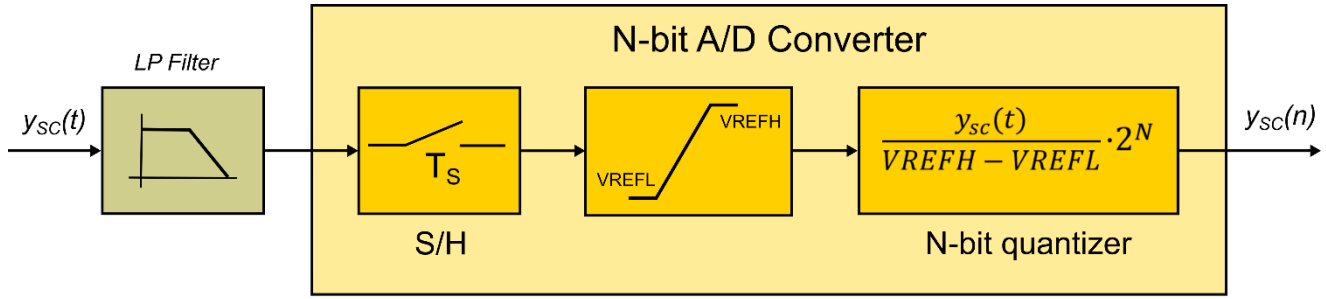
- Anti-aliasing filter
  - Low pass analog filter
  - It eliminates all frequencies out of our interest
  - It is always recommended to use anti-aliasing filter in order to eliminate noise
  - Design filter to eliminate all frequencies above $\frac{\omega_S}{2}$

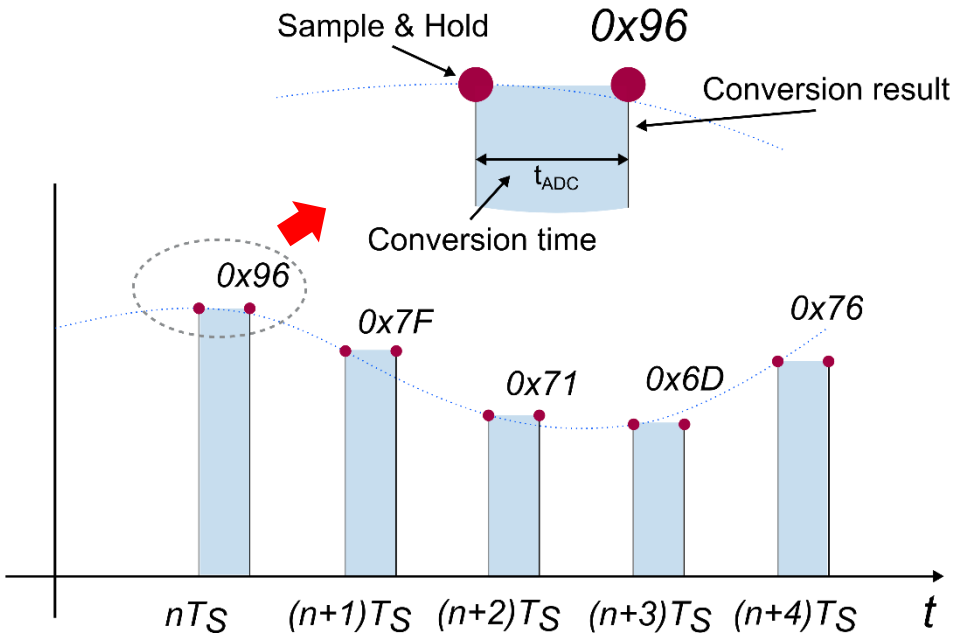




Digital Control Loop Design – Sampling

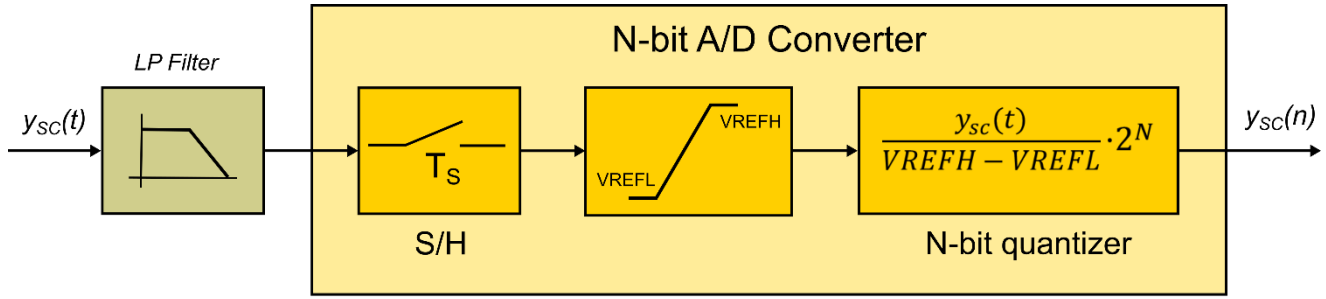
- A/D Converter
  - ADC consists of Sample&Hold circuit and quantizer. At first the Sample&Hold circuit samples input voltage. Subsequently the quantizer converts sample to the number
  - Both Sample&Hold and conversion to the number takes time known as ADC conversion time.

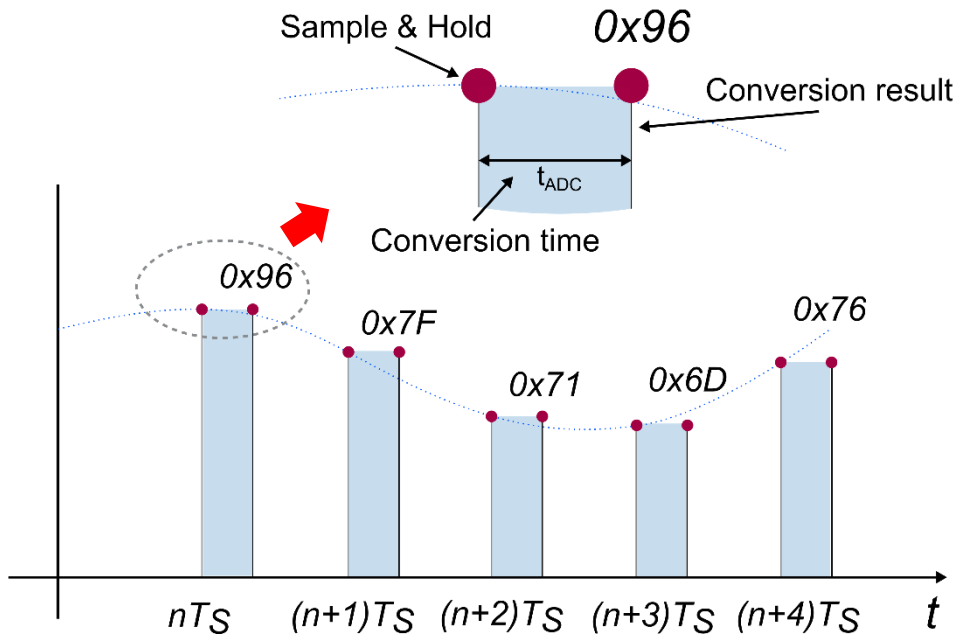




Digital Control Loop Design – Sampling

- A/D Converter
  - Both sample and conversion times are usually user configurable
  - The faster ADC ⇒ the better, but
    - Faster ADC is more expensive
    - Faster ADC ⇒ higher consumption
    - Fast sampling (charging sampling & hold capacitor) ⇒ low input impedance ⇒ expensive
    - Compromise has to be find



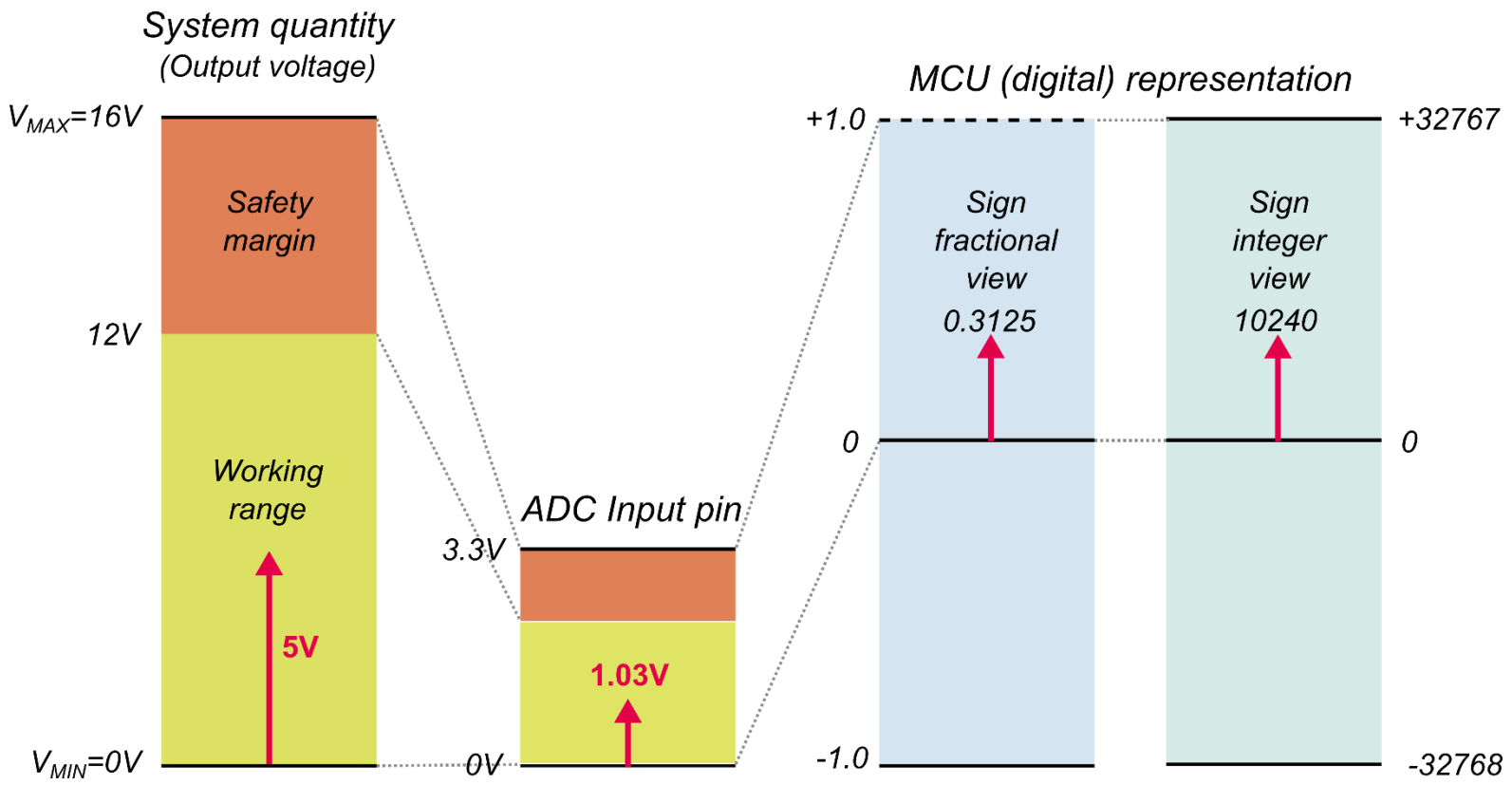


Digital Control Loop Design – Sampling

- A/D Converter
  - Scaling quantities
    - The NXP uses fractional arithmetic for algorithms calculation.
    - Also some NXP ADCs supports fractional format
    - The ADC gain can be expressed as

$$K_{ADC} = \frac{1}{Y_{MAX}}$$

Where Y_{MAX} is measurable input range of measured quantity



Digital Control Loop Design – z Transform

We can rewrite $y'(t)$ as

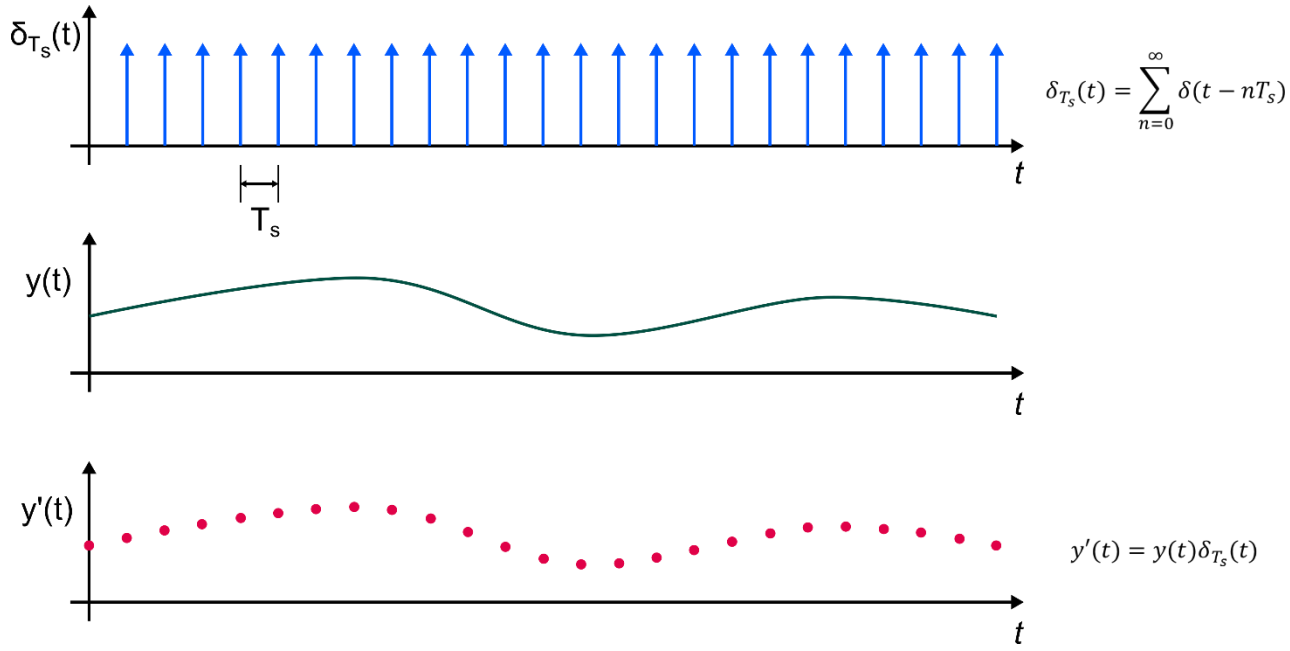
$$y'(t) = \sum_{n=0}^{\infty} y(nT)\delta(t - nT)$$

The Laplace transform of $y'(t)$ is

$$Y'(s) = \sum_{n=0}^{\infty} y(nT)e^{-snT}$$

Defining $z = e^{sT}$, we can write

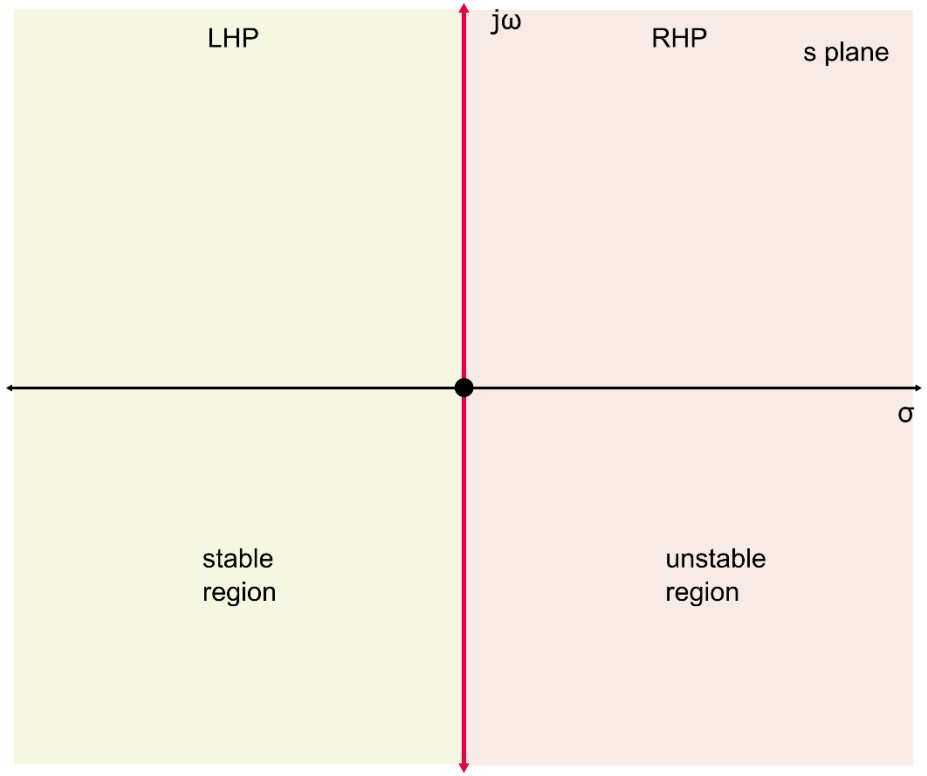
$$Y'(z) = \sum_{n=0}^{\infty} y(nT)z^{-n}$$

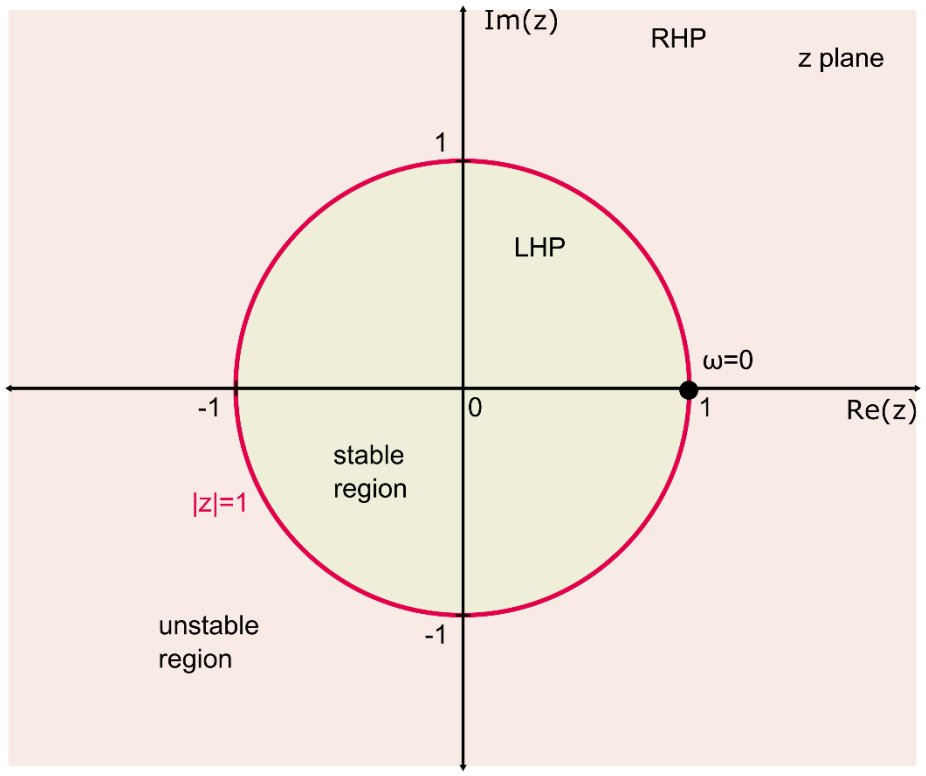


Digital Control Loop Design – z Transform

- Mapping of s-plane into z-plane

$$z = e^{sT_s} = e^{(\sigma + j\omega)T_s} = e^{\sigma T_s} e^{j\omega T_s} = |z| e^{j\theta}$$

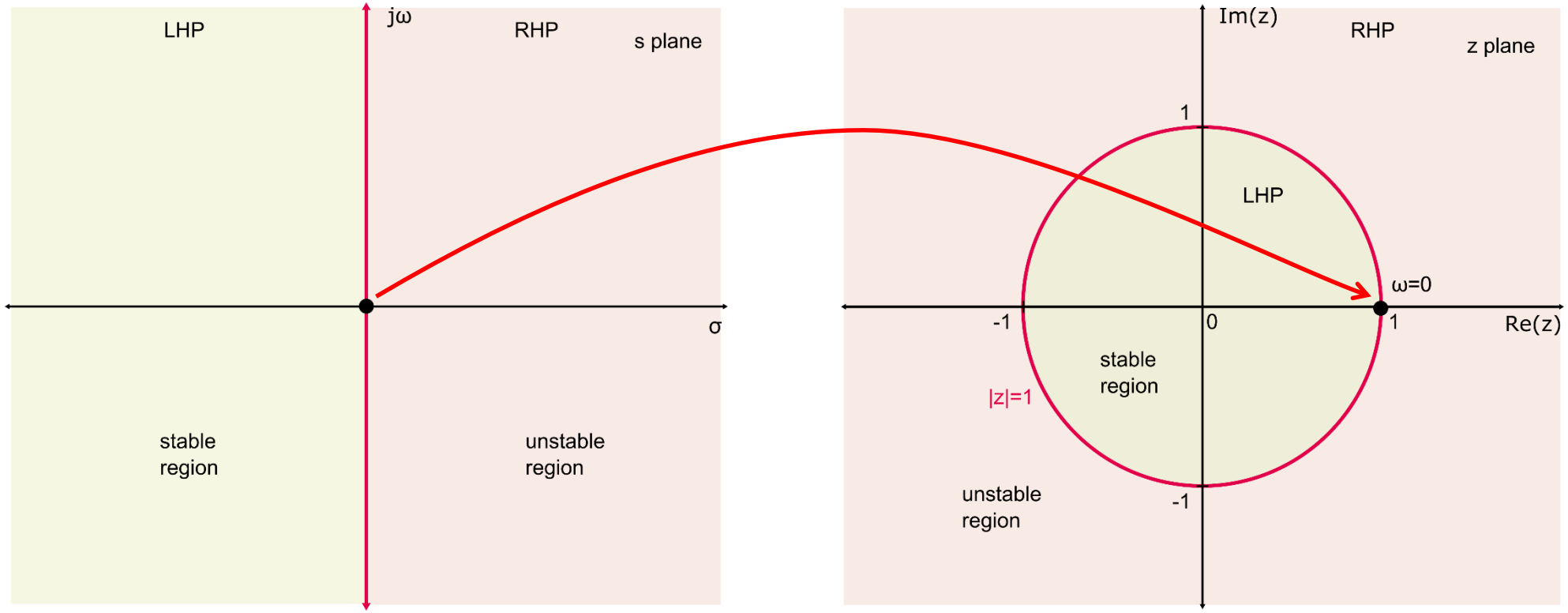




Digital Control Loop Design – z Transform

- Mapping of s-plane into z-plane

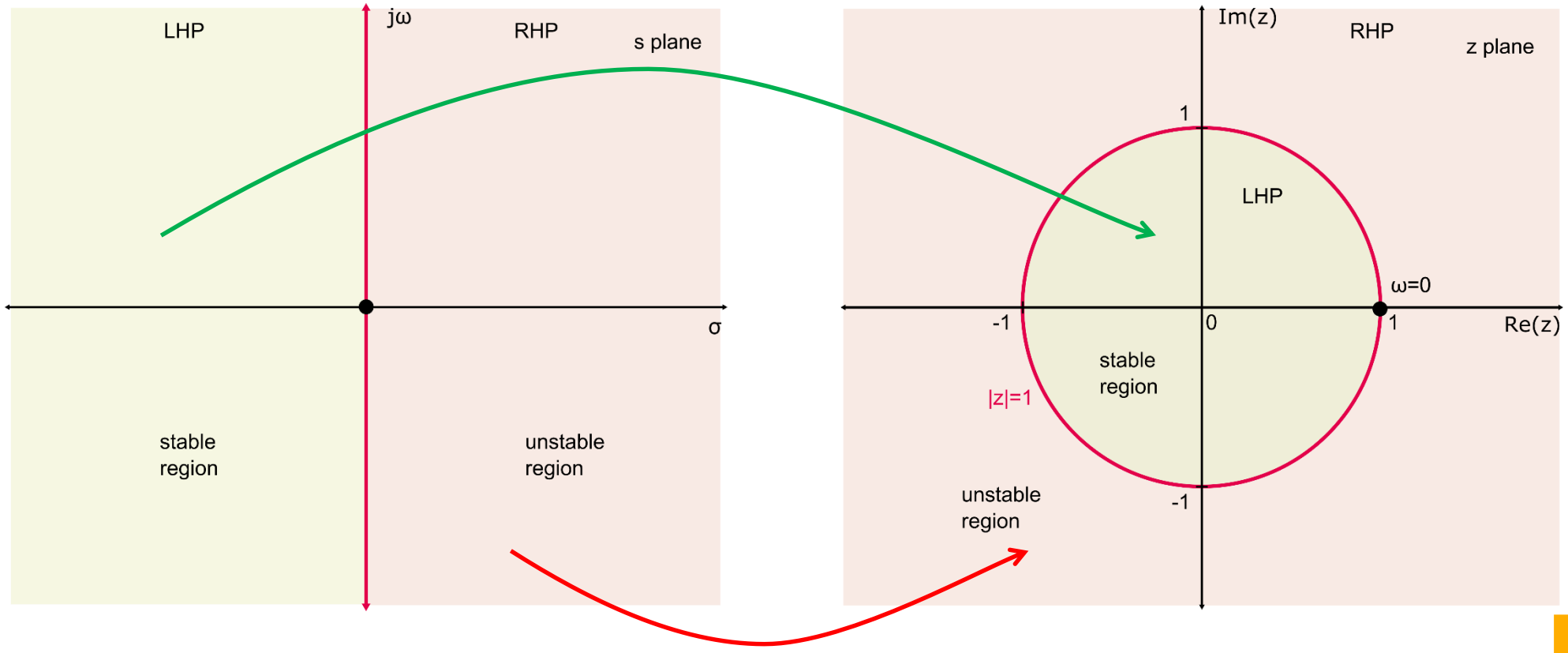
For $\sigma, \omega = 0 \Rightarrow z = e^{0 \cdot T_s}\ e^{j0 \cdot T_s} = 1$



Digital Control Loop Design – z Transform

- Mapping of s-plane into z-plane

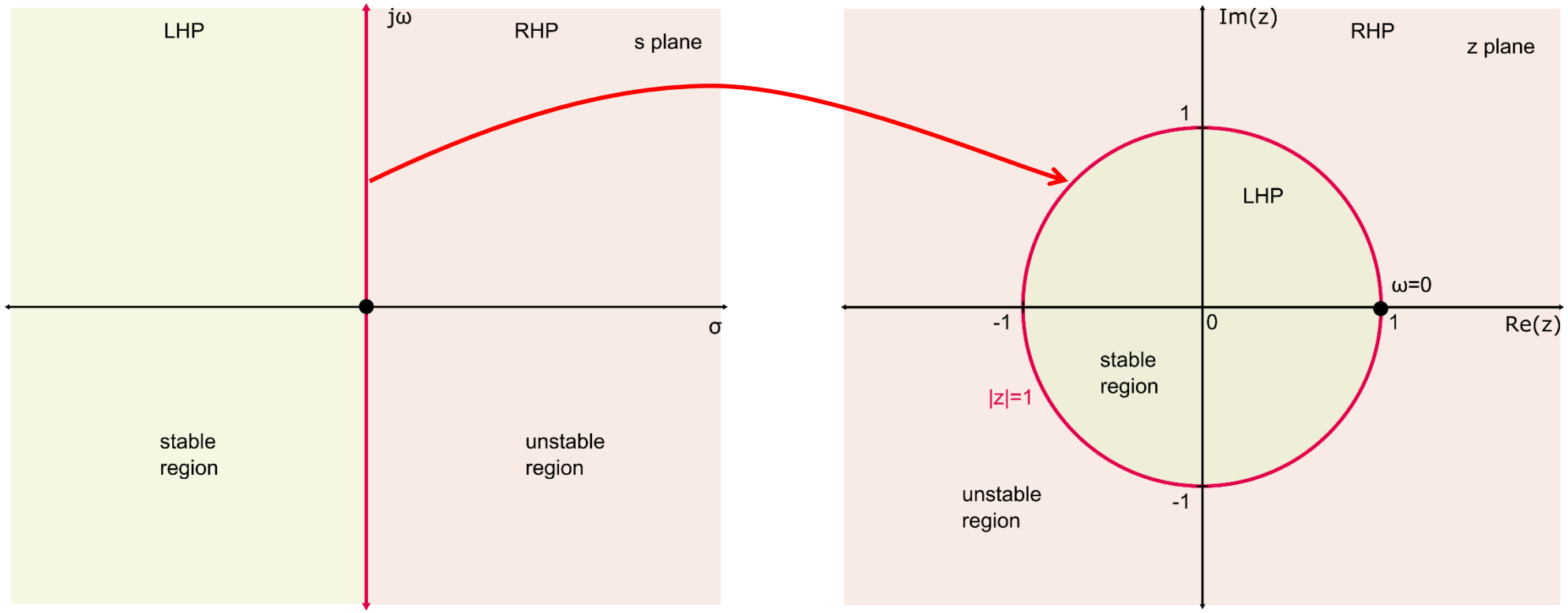
For $\sigma < 0 \Rightarrow |z| = e^{\sigma \cdot T_s} < 1$, for $\sigma > 0 \Rightarrow |z| = e^{\sigma \cdot T_s} > 1$



Digital Control Loop Design – z Transform

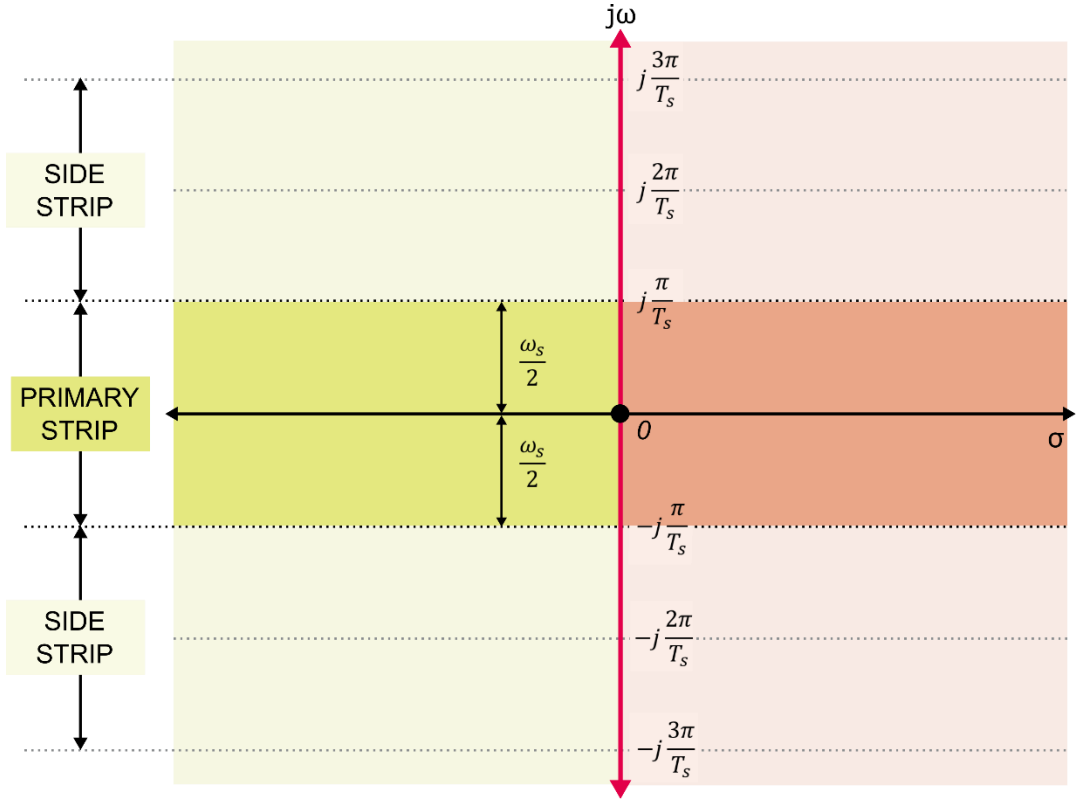
- Mapping of s-plane into z-plane

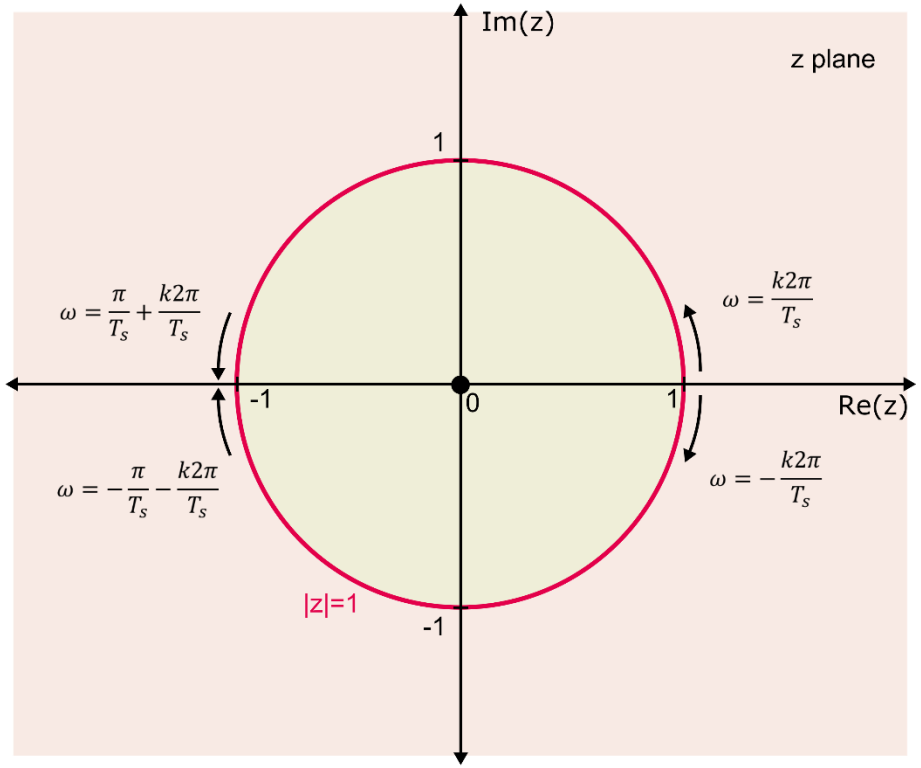
For $\sigma = 0 \Rightarrow |z| = e^{0 \cdot T_s} = 1, e^{j\omega T_s}$



Digital Control Loop Design – z Transform

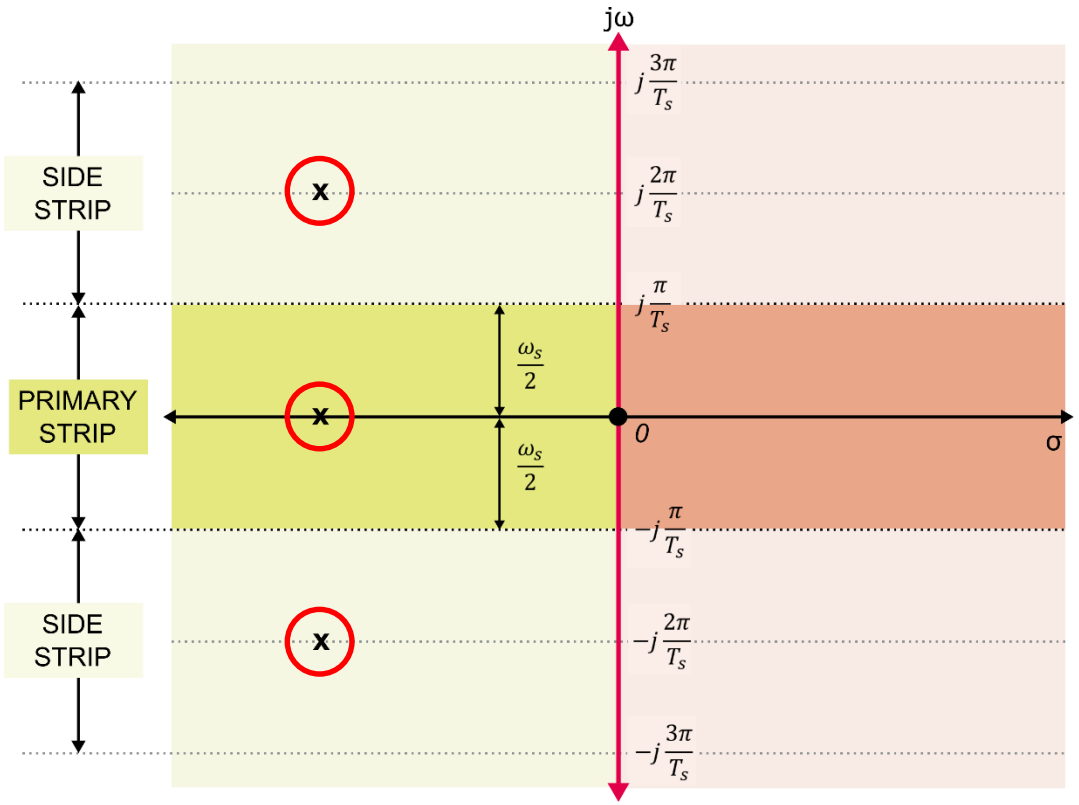
- Periodicity of s-plane into z-plane mapping
  - Because z is periodic ($z = e^{sT_s \pm jk2\pi}$), the multiple zeros and poles in s-domain map into single point in z-domain (aliasing)

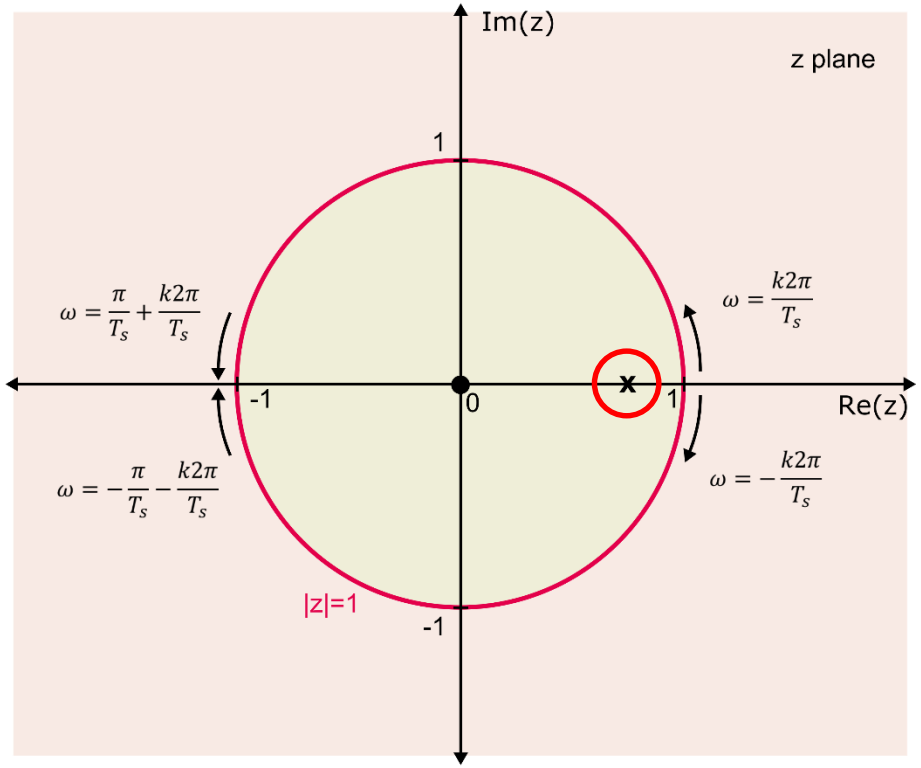




Digital Control Loop Design – z Transform

- Periodicity of s-plane into z-plane mapping
  - All three poles in s-plane map as single pole in z-plane





Digital Control Loop Design – z Transform

- Transform Methods from s to z domain
  - Backward Difference
  - Forward Difference
  - Bilinear (Tustin) Transformation

  } Integral Transformations

  - Step Invariance (ZOH)
  - First Invariance (FOH)
  - Matched pole-zero mapping

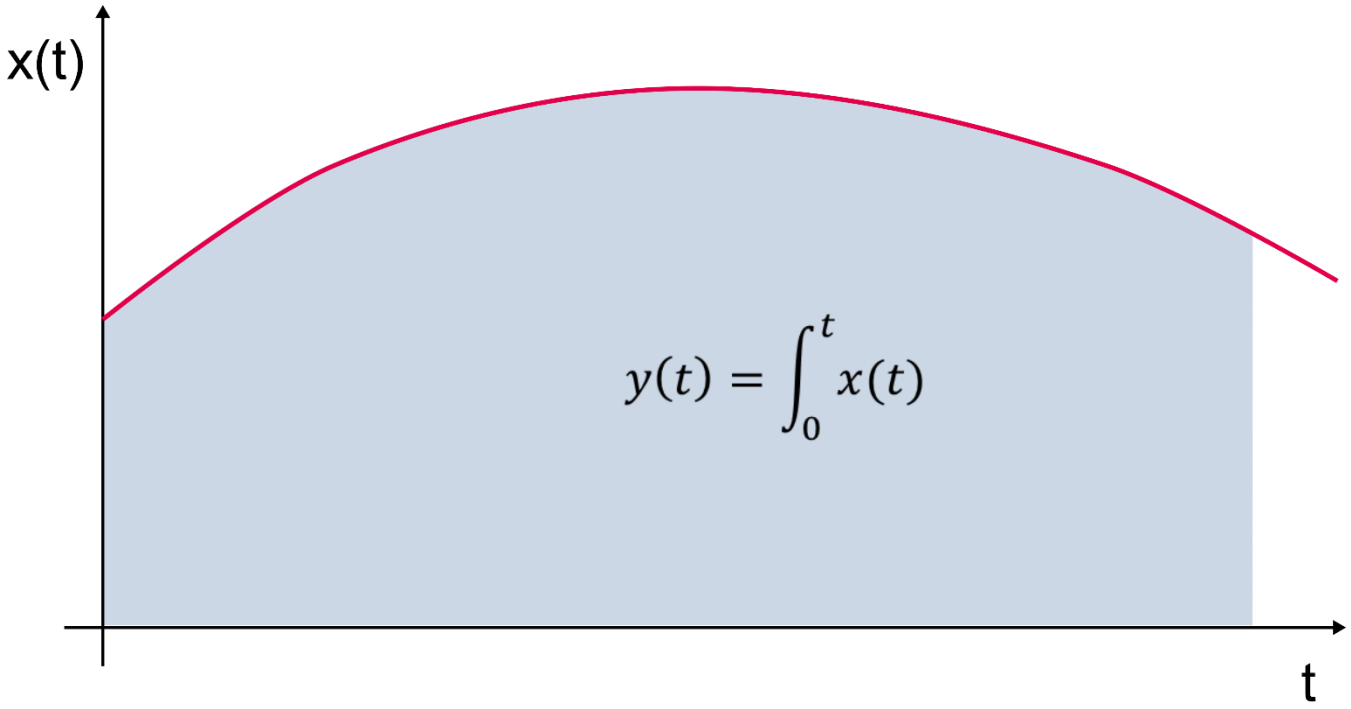
Digital Control Loop Design – z Transform

- Integral Transformations
  - Let's have integral function

$$y(t) = \int_0^t x(t)$$

  - The Laplace transform of y(t) is

$$Y(s) = \frac{1}{s}\mathcal{L}\{x(t)\}$$



Digital Control Loop Design – z Transform

- Integral Transformations
  - Now we approximate integral as

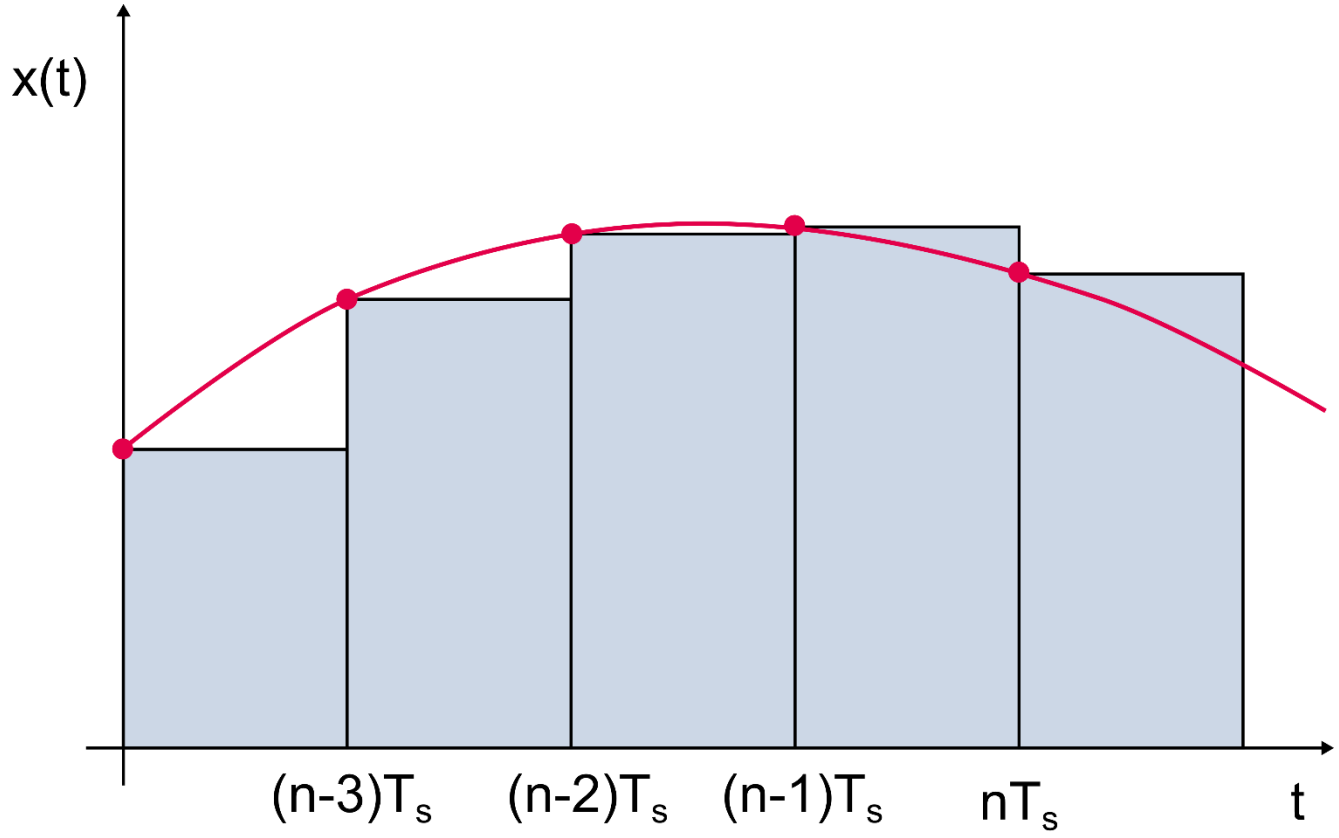
$$\int_0^t x(t) \approx y[n] = y[n-1] + x[n-1] \cdot T_s$$

  - The z-transform of this equation is

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

  After rearranging we get

$$Y(z) = \left(\frac{T_s}{z-1}\right) X(z)$$



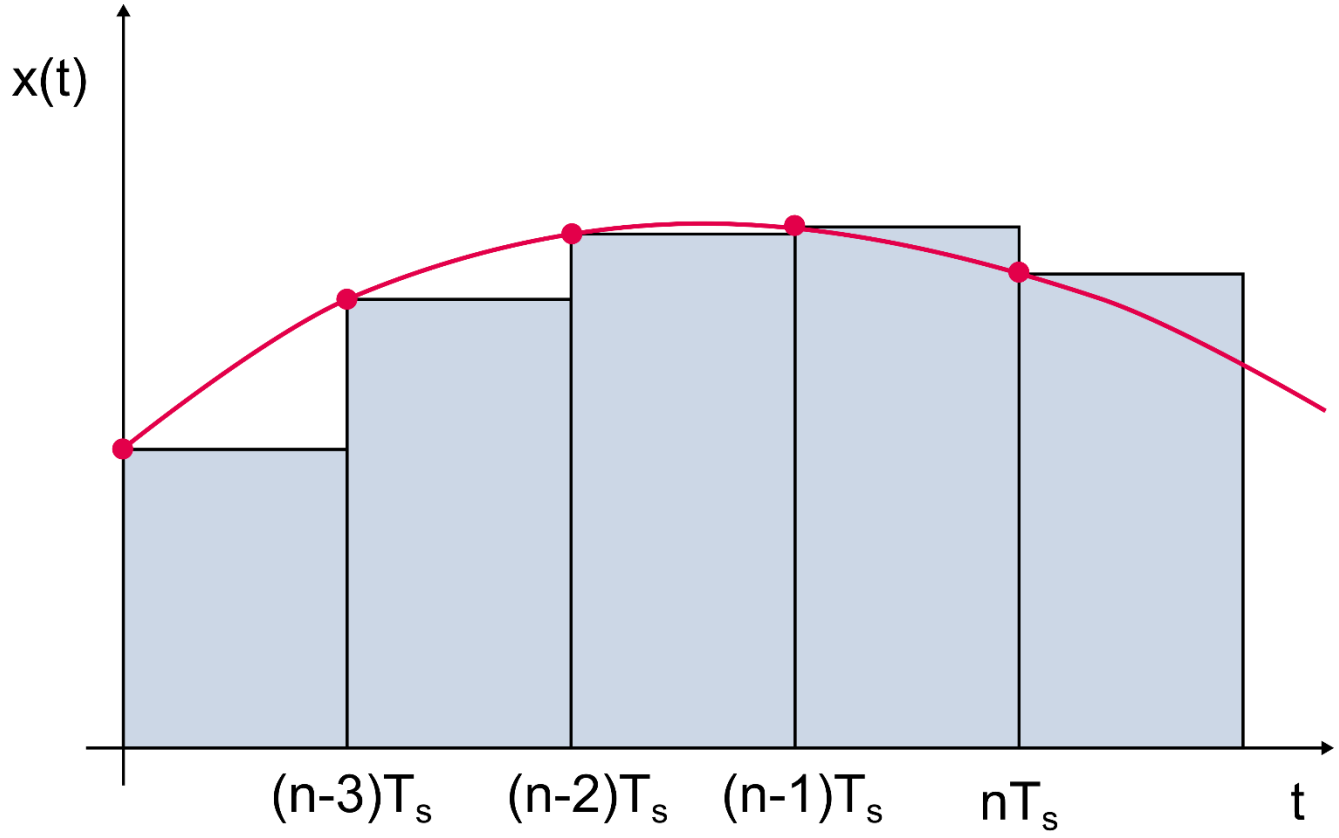
Digital Control Loop Design – z Transform

- Integral Transformations
  - Comparing with

$$Y(s) = \frac{1}{s} X(s)$$

  - We can write

$$\frac{1}{s} \approx \left(\frac{T_s}{z-1} \right)$$



Digital Control Loop Design – z Transform

- Integral Transformations
  - Comparing with

$$Y(s) = \frac{1}{s} X(s)$$

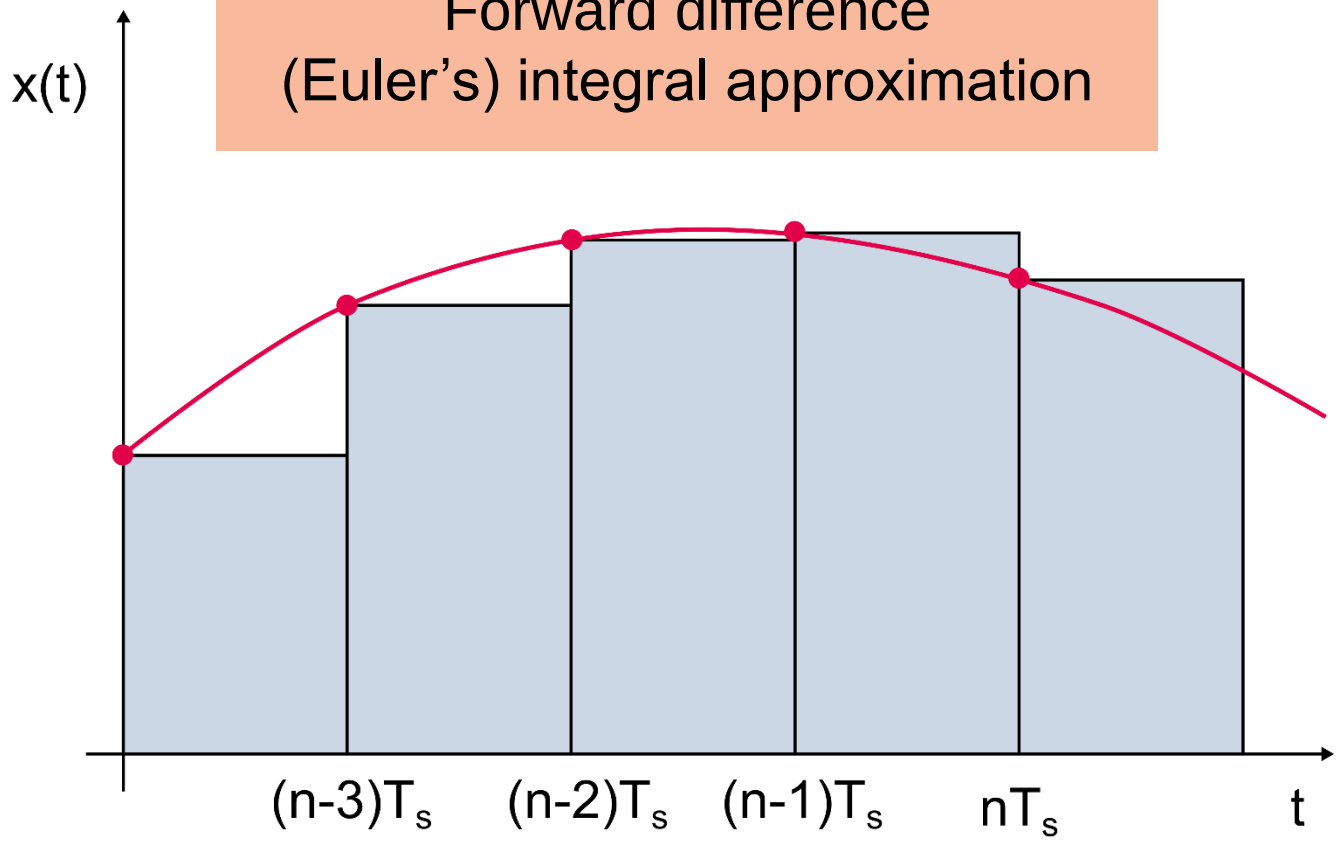
  - We can write

$$\frac{1}{s} \approx \left(\frac{T_s}{z-1} \right)$$

or

$$\boldsymbol{s \approx \frac{z-1}{T_s}}$$

Forward difference
(Euler's) integral approximation



Digital Control Loop Design – z Transform

- Integral Transformations
  - Now we approximate integral as

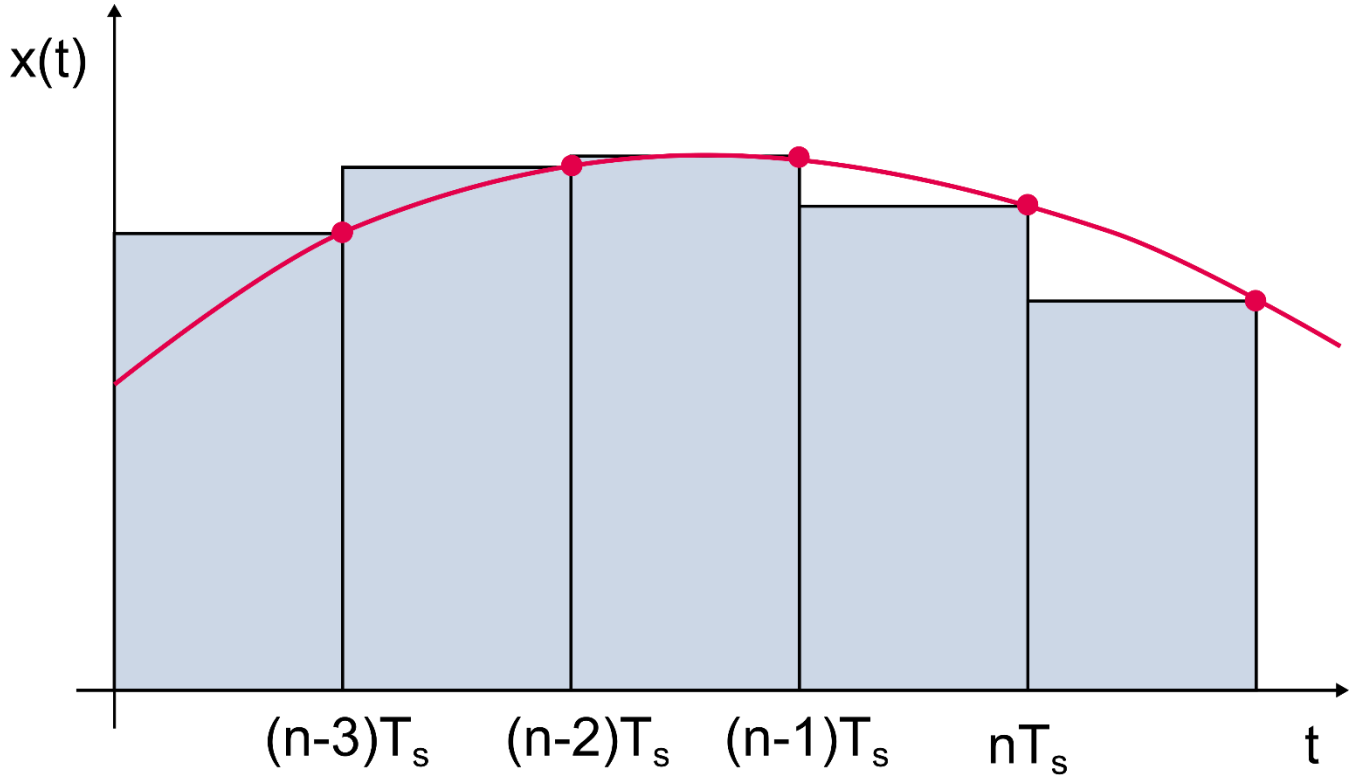
$$\int_0^t x(t) \approx y[n] = y[n-1] + x[n] \cdot T_s$$

  - The z-transform of this equation is

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

  After rearranging we get

$$Y(z) = \left(\frac{T_s z}{z-1}\right) X(z)$$



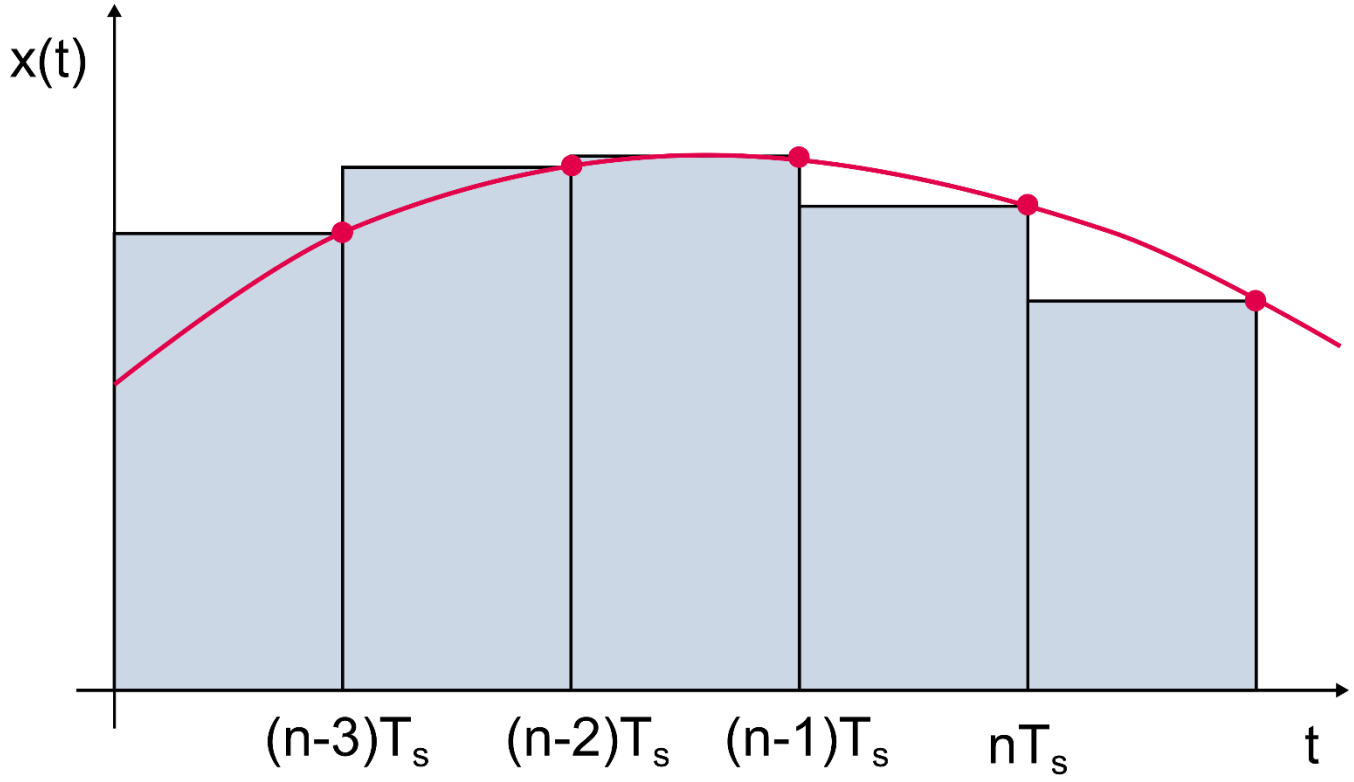
Digital Control Loop Design – z Transform

- Integral Transformations
  - Comparing with

$$Y(s) = \frac{1}{s} X(s)$$

  - We can write

$$\frac{1}{s} \approx \left(\frac{T_s z}{z-1} \right)$$



Digital Control Loop Design – z Transform

- Integral Transformations
  - Comparing with

$$Y(s) = \frac{1}{s} X(s)$$

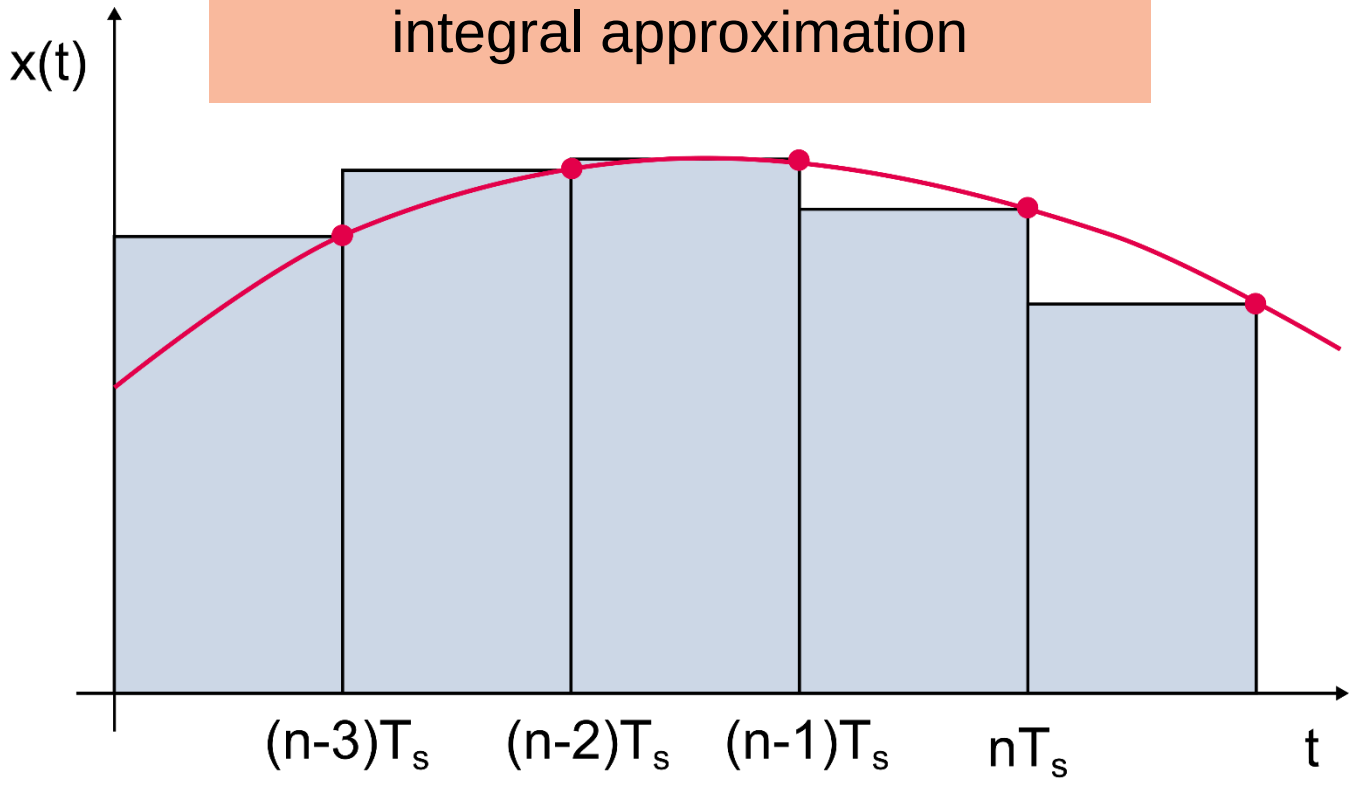
  - We can write

$$\frac{1}{s} \approx \left(\frac{T_s z}{z-1} \right)$$

or

$$\boldsymbol{s \approx \frac{z-1}{T_s z}}$$

Backward difference integral approximation



Digital Control Loop Design – z Transform

- Integral Transformations
  - Now we approximate integral as

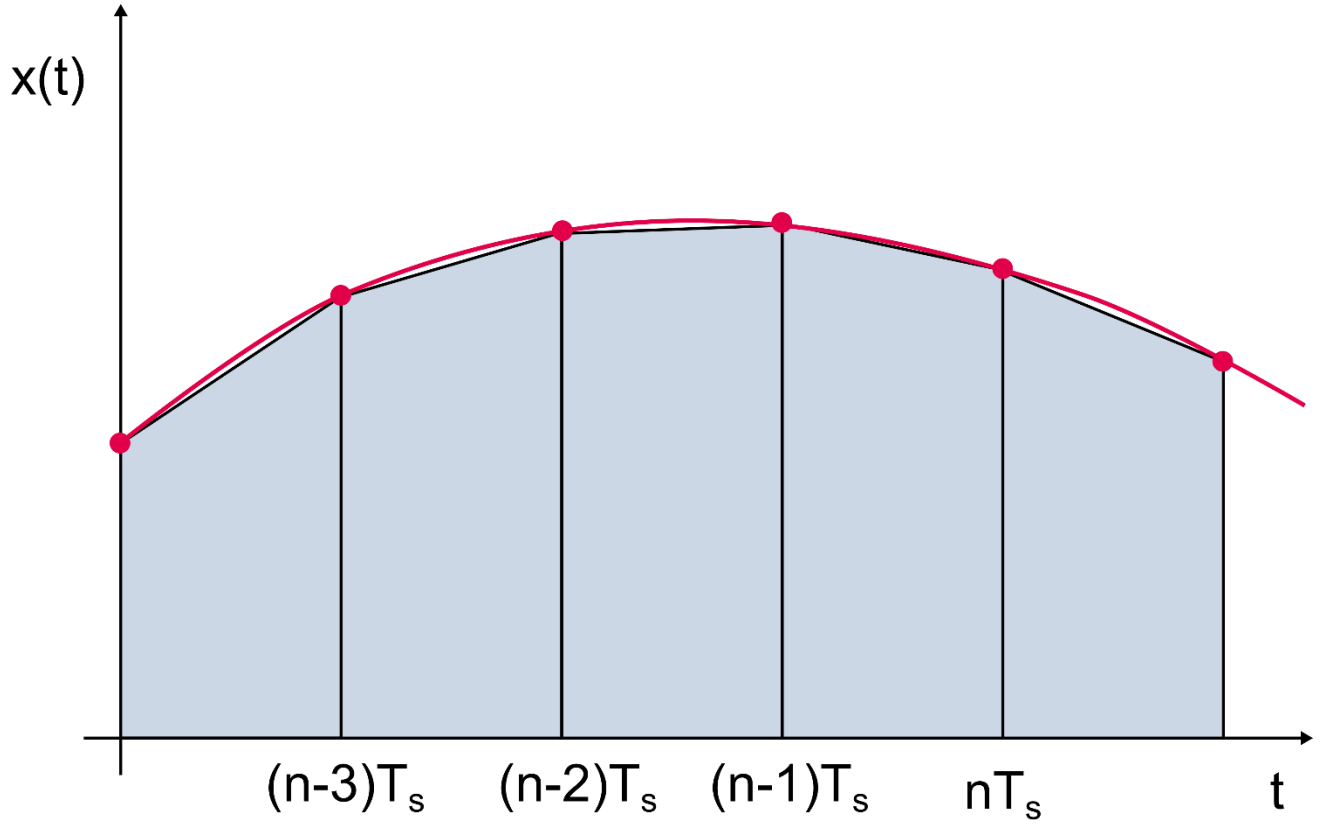
$$y[n] = y[n-1] + \frac{x[n] + x[n-1]}{2} \cdot T_s$$

  - The z-transform of this equation is

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

  After rearranging we get

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



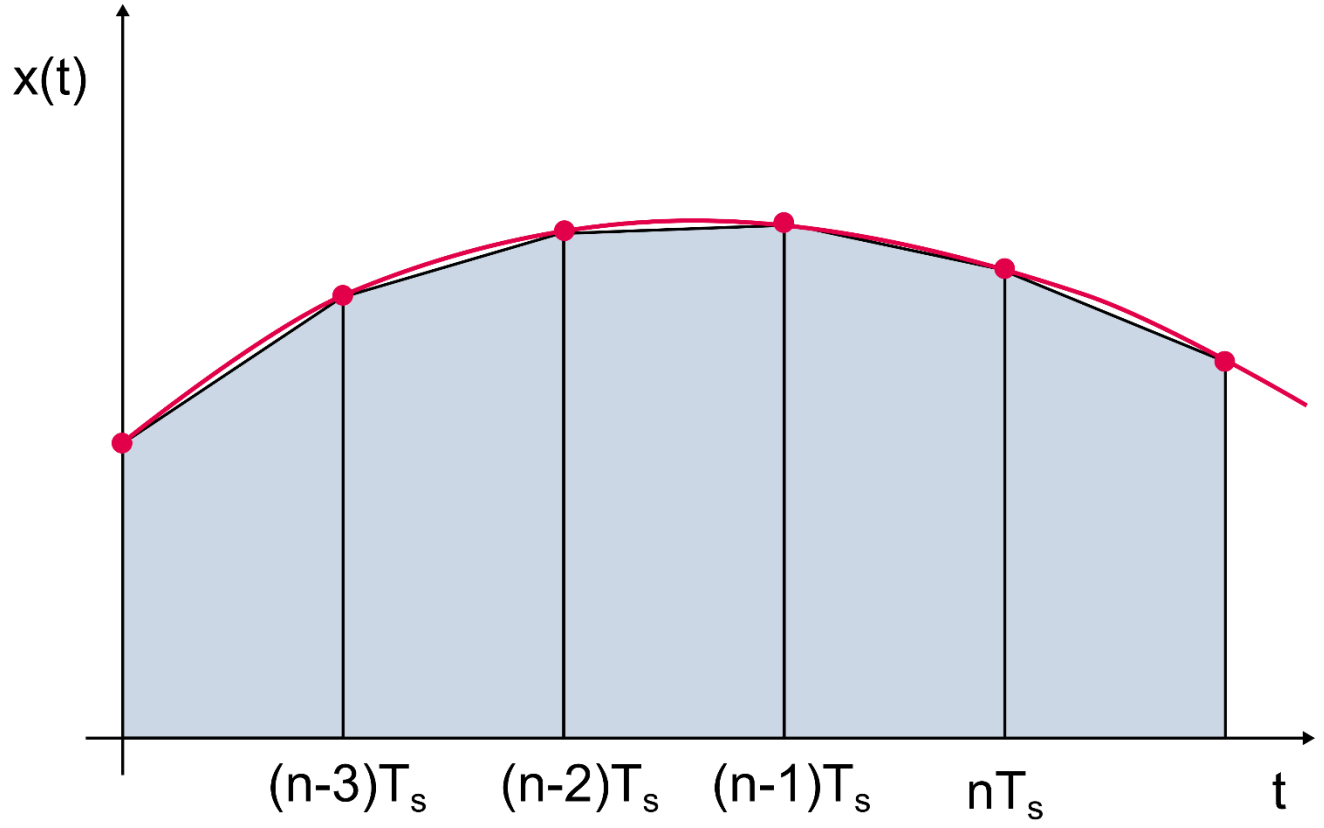
Digital Control Loop Design – z Transform

- Integral Transformations
  - Comparing with

$$Y(s) = \frac{1}{s} X(s)$$

  - We can write

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



Digital Control Loop Design – z Transform

- Integral Transformations
  - Comparing with

$$Y(s) = \frac{1}{s} X(s)$$

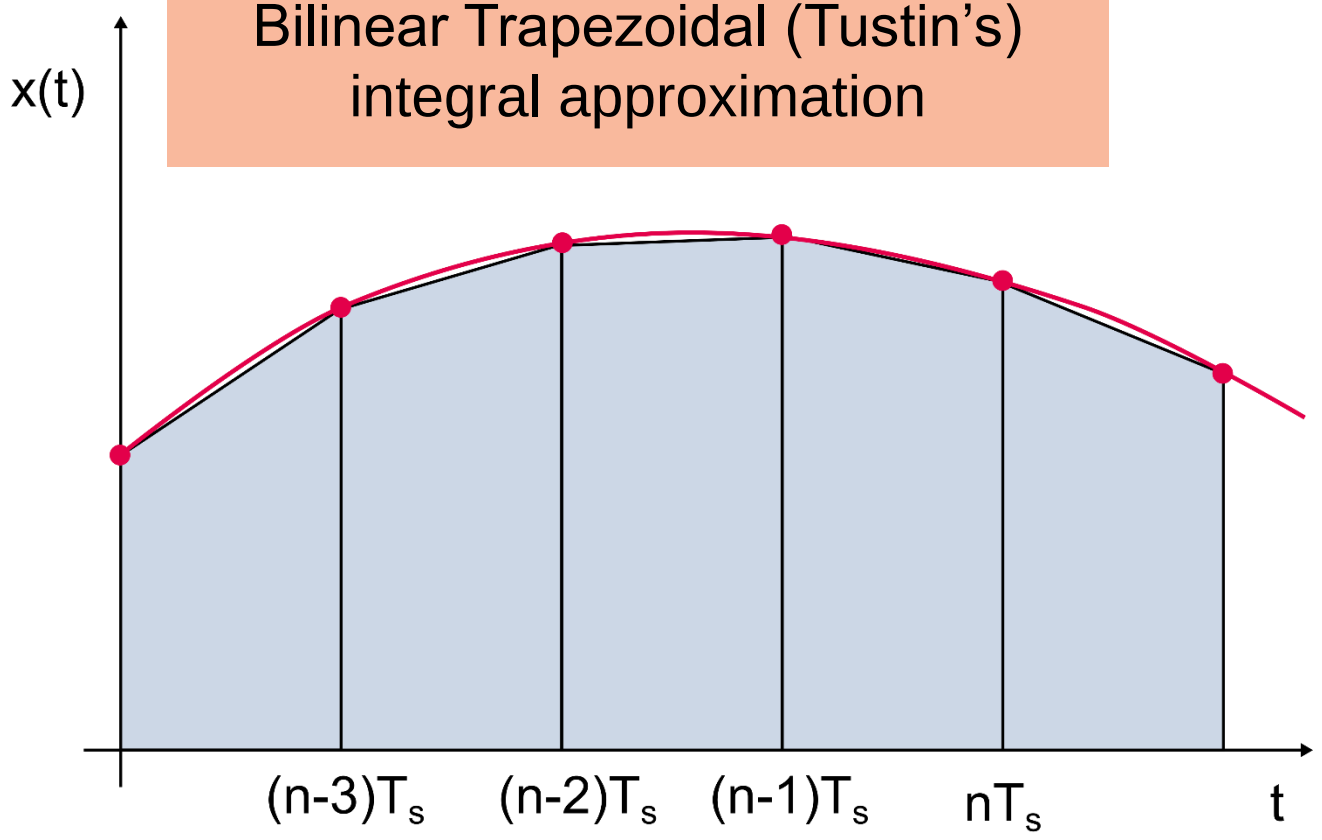
  - We can write

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

or

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

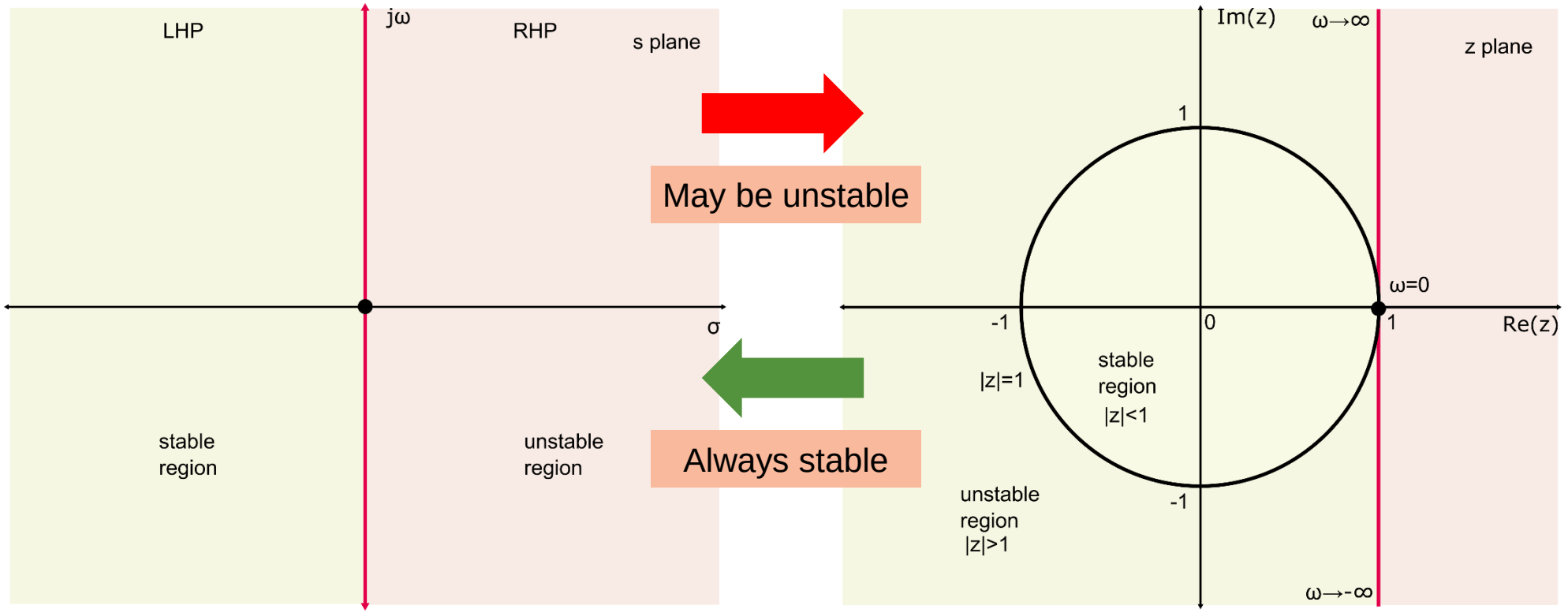
Bilinear Trapezoidal (Tustin's) integral approximation



Digital Control Loop Design – z Transform

- Integral Transformations -Forward (Euler's) difference mapping

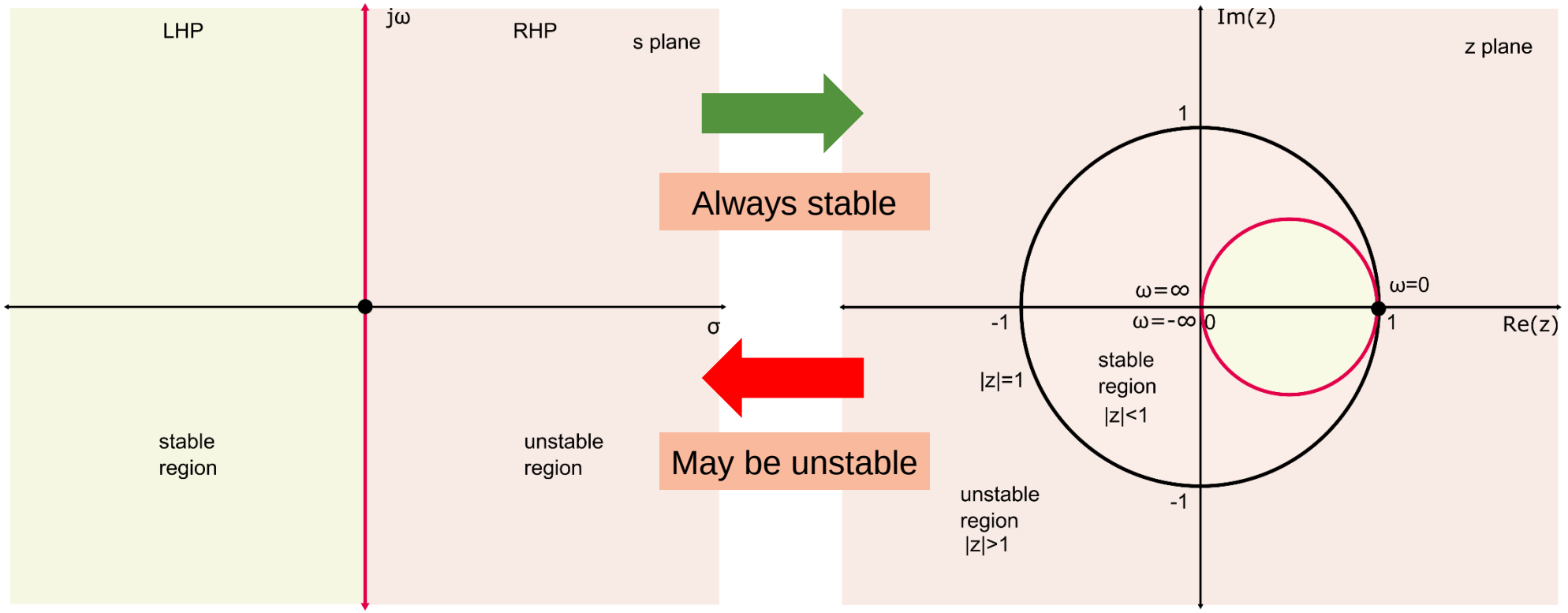
$$s \approx \frac{z-1}{T_s} \quad \Rightarrow \quad z \approx 1 + sT_s$$



Digital Control Loop Design – z Transform

- Integral Transformations -Backward difference mapping

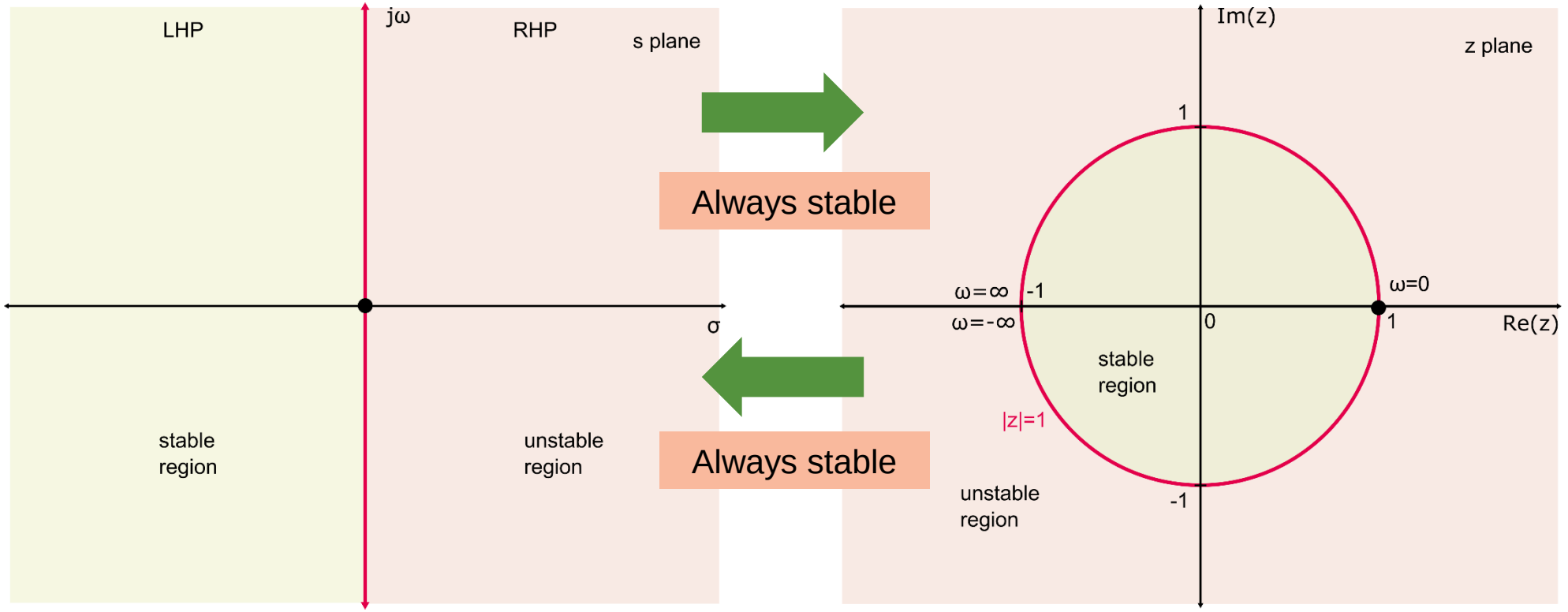
$$s \approx \frac{z-1}{T_s z} \quad \Rightarrow \quad z \approx \frac{1}{1-sT_s}$$



Digital Control Loop Design – z Transform

- Integral Transformations – Bilinear (Tustin's) mapping

$$s \approx \frac{2}{T_s}\frac{z-1}{z+1} \quad \Rightarrow \quad z \approx \frac{1+sT_s/2}{1-sT_s/2}$$



Digital Control Loop Design – z Transform

- Integral Transformations – Bilinear (Tustin's mapping – Frequency warping
  - Let's substitute $z = e^{\sigma T_s} e^{j\omega T_s}, s = \sigma + j\omega_a, \sigma = 0$ into

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

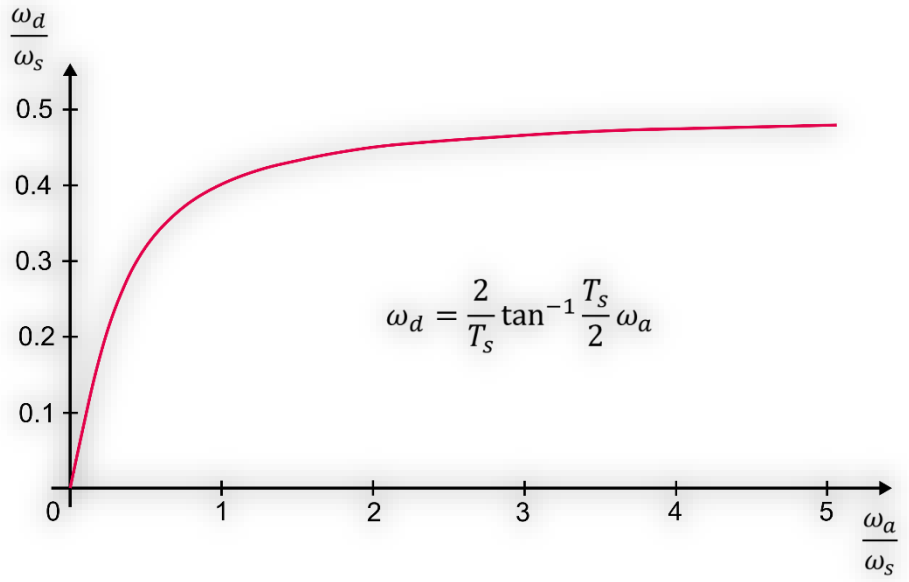
Then we can write

$$j\omega_a = \frac{2}{T_s}\frac{e^{j\omega_d T_s} - 1}{e^{j\omega_d T_s} + 1}$$

$$\omega_a = \frac{2}{T_s}\tan\frac{\omega_d T_s}{2}$$

And finally

$$\omega_d = \frac{2}{T_s}\tan^{-1}\frac{T_s}{2}\omega_a$$



There is frequency distortion
When Tustin's mapping is used

Digital Control Loop Design – z Transform

- Integral Transformations – Bilinear (Tustin) mapping – Frequency Pre-warping
  - In order to prevent distortion the so called prewarping has to be done before s to z mapping

$$\omega_a^{prewarped} = \frac{2}{T_s}\tan\frac{\omega_a T_s}{2}$$

  - The bilinear mapping with prewarping is done by following steps:
    1. Find all zeros and poles frequencies in s-domain
    2. Calculate theirs prewarped equivalents
    3. Use prewarped frequencies for s to z bilinear transformation

$$\omega_a \Rightarrow \omega_a^{prewarped} \Rightarrow \omega_d$$

Digital Control Loop Design – z Transform

- Integral Transformations – Bilinear (Tustin) mapping – Frequency Prewarping
  - Do I need to use prewarping?
    - If zero or pole frequency is far below sampling frequency, the prewarping can be neglected for controller design since the system is usually sampled 10-times faster for proper response
    - The prewarping is usually used for filter design in order to have exact frequency characteristic

$\frac{\omega_a}{\omega_s}$	$\frac{\omega_d}{\omega_s}$	*Error [%]*
0.05	0.049	0.8
0.1	0.097	3.1
0.2	0.179	10.1
0.3	0.241	19.8
0.4	0.286	28.5
0.5	0.320	36.1

Digital Control Loop Design – z Transform

- Difference equation
  - Applying selected s to z transform we get transfer function in z-domain

$$\frac{y(z)}{x(z)} = \frac{B_0 z^m + B_1 z^{m-1} + \cdots + B_{m-1}\, z + A_m}{A_0 z^n + A_1 z^{n-1} + \cdots + A_{n-1}\, z + A_n}$$

  - After rearranging into normalized form we get

$$\frac{y(z)}{x(z)} = \frac{b_0 z^{m-n} + b_1 z^{m-1-n} + \cdots + b_{m-1}\, z^{1-n} + b_0 z^{-n}}{1 + a_1 z^1 + \ldots + a_{n-1}\, z^{1-n} + a_n z^{-n}}$$

  - Considering that z^{-1} means value in step k-1 we can write

$$y(k) = b_0 x(k-n+m) + b_1 x(k-n+m+1) + .. + b_{m-1} x(k-n+1) + b_m x(k-n)$$
$$-a_1 y(k-1) - \ldots - a_{n-1} y(k-2) - a_n y(k-n)$$

Digital Control Loop Design – z Transform

- Example
  - Discretize Type II controller from previous buck example using Tustin approximation and sampling time $T_s = 5\ \mu s$

$$G_C(s) = \frac{u(s)}{e(s)} = \frac{3781584(s + 7532)}{s\ (s + 6.283 \cdot 10^5)}$$

  - Let's replace $\boldsymbol{s \Longrightarrow \frac{2}{T_s}\frac{z-1}{z+1}}$ and after rearranging we get

$$G_C(z) = \frac{u(z)}{e(z)} = \frac{3.7467(z + 1)(z - 0.963)}{(z - 1)(z - 0.222)} = \frac{3.747 + 0.1385z^{-1} - 3.608z^{-2}}{1 - 0.778z^{-1} - 0.222z^{-2}}$$

  - And in difference form

$$u(k) = 0.778u(k - 1) + 0.222u(k - 2) + 3.747e(k) + 0.1385e(k - 1) - 3.608e(k - 2)$$

Digital Control Loop Design – z Transform

- Example
  - Discretize Type II controller from previous buck example using Tustin approximation and sampling time $T_s = 5\ \mu s$

$$G_C(s) = \frac{u(s)}{e(s)} = \frac{3781584(s + 7532)}{s\ (s + 6.283 \cdot 10^5)}$$

  - Let's replace $\boldsymbol{s \Longrightarrow \frac{2}{T_s}\frac{z-1}{z+1}}$ and after rearranging we get

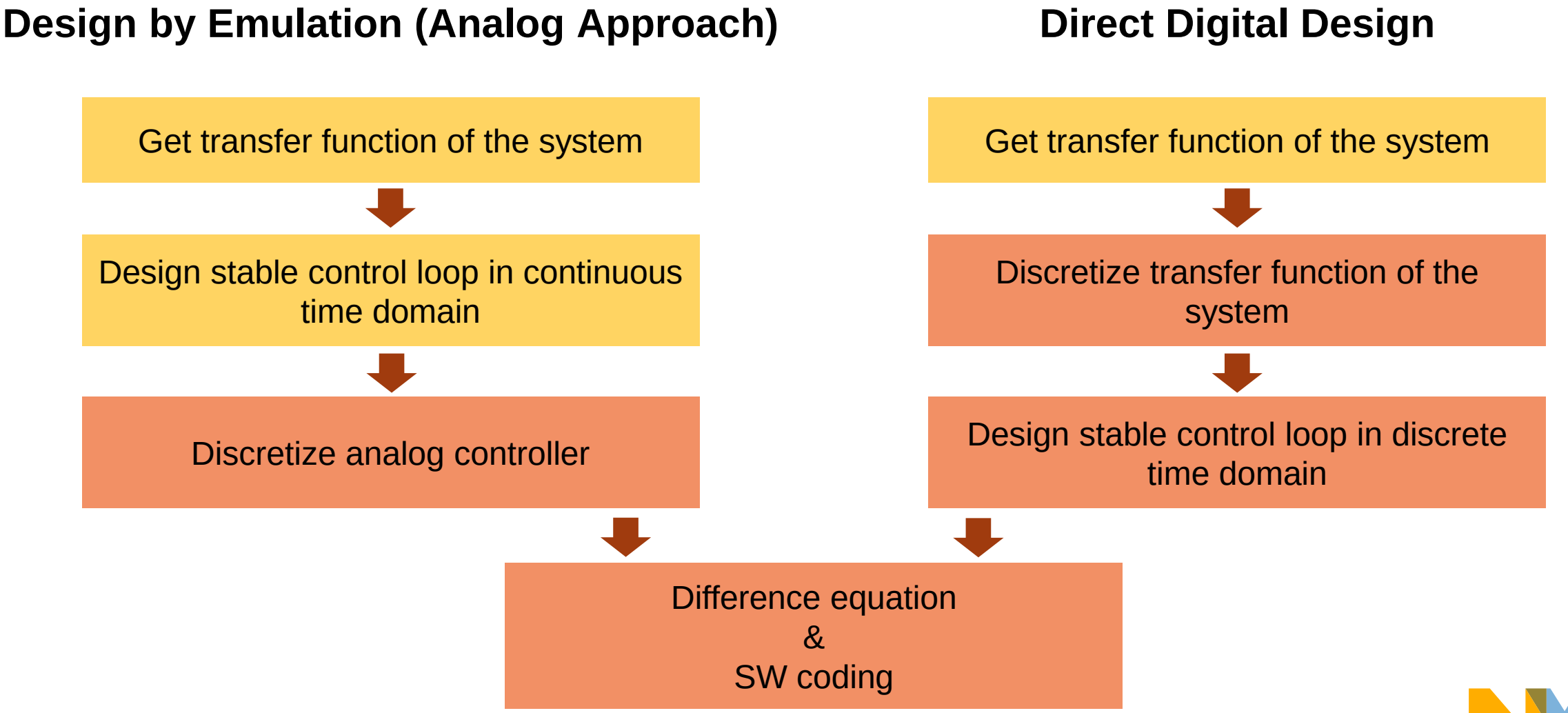
$$G_C(z) = \frac{u(z)}{e(z)} = \frac{3.7467(z + 1)(z - 0.963)}{(z - 1)(z - 0.222)} = \frac{3.747 + 0.1385z^{-1} - 3.608z^{-2}}{1 - 0.778z^{-1} - 0.222z^{-2}}$$

  - And in difference form

$$u(k) = 0.778u(k - 1) + 0.222u(k - 2) + 3.747e(k) + 0.1385e(k - 1) - 3.608e(k - 2)$$

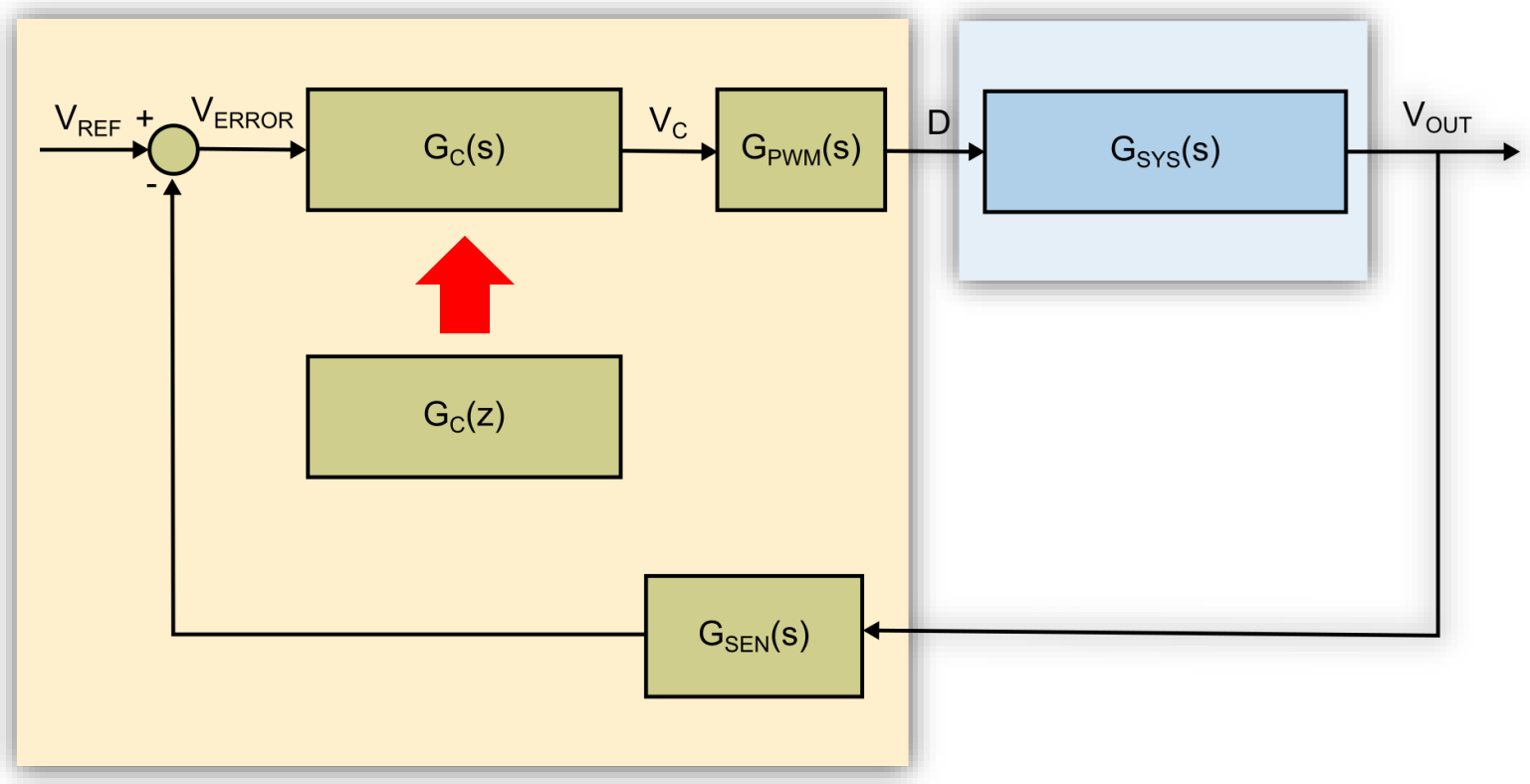
Please note that analog controller **Type II** is transformed into **2P2Z** digital equivalent

Digital Control Loop Design – Designing the Control Loop



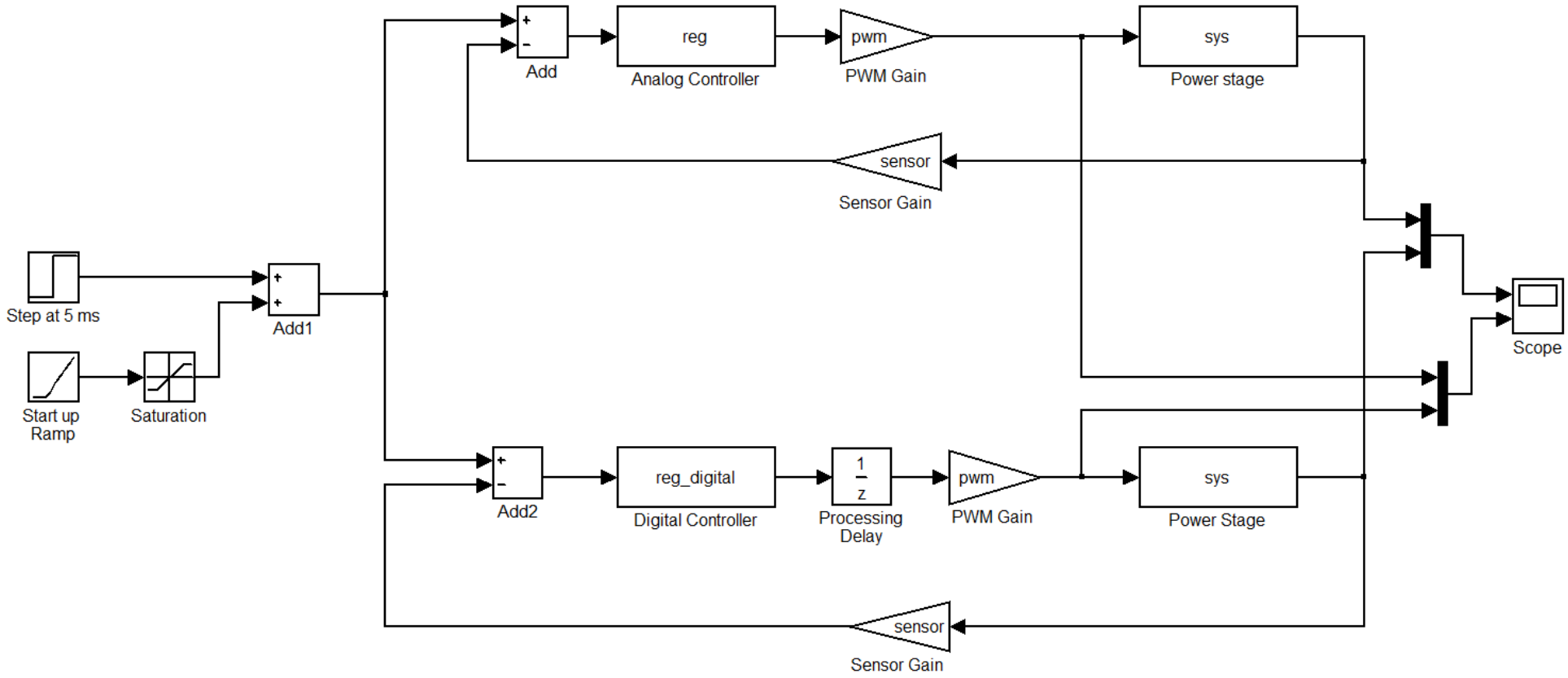
Digital Control Loop Design – Designing the Control Loop

- Control loop design by analog approach



Digital Control Loop Design – Designing the Control Loop

- MATLAB Model for analog and digital control loop comparison

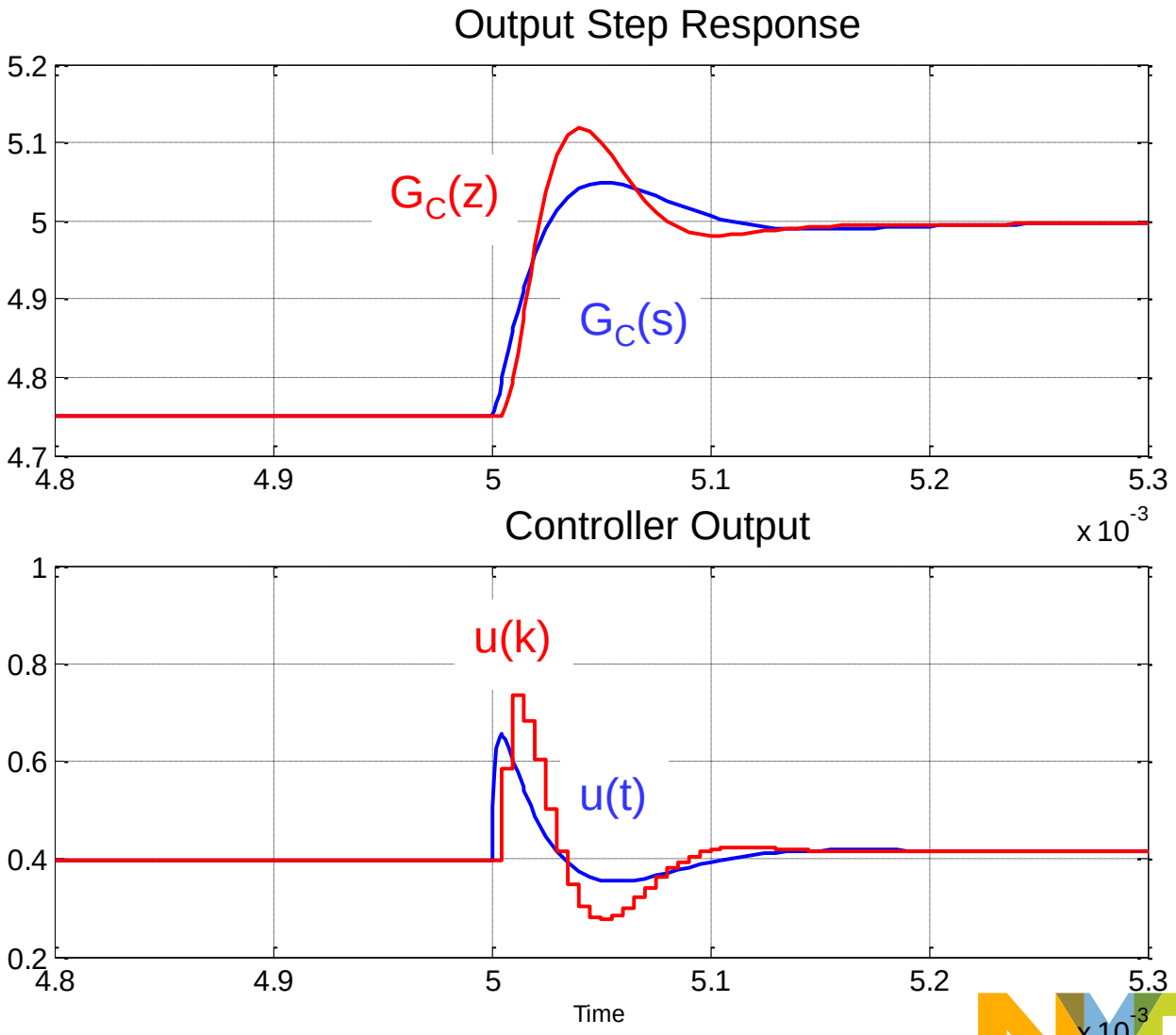


Digital Control Loop Design – Designing the Control Loop

- $G_C(s)$ replaced by $G_C(z)$, $T_s = 5\ \mu s$

$$G_C(z) = \frac{3.747 + 0.1385z^{-1} - 3.608z^{-2}}{1 - 0.778z^{-1} - 0.222z^{-2}}$$

- We can see oscillations and larger overshoot, but the system is still stable. The phase margin is significantly smaller than at analog controller

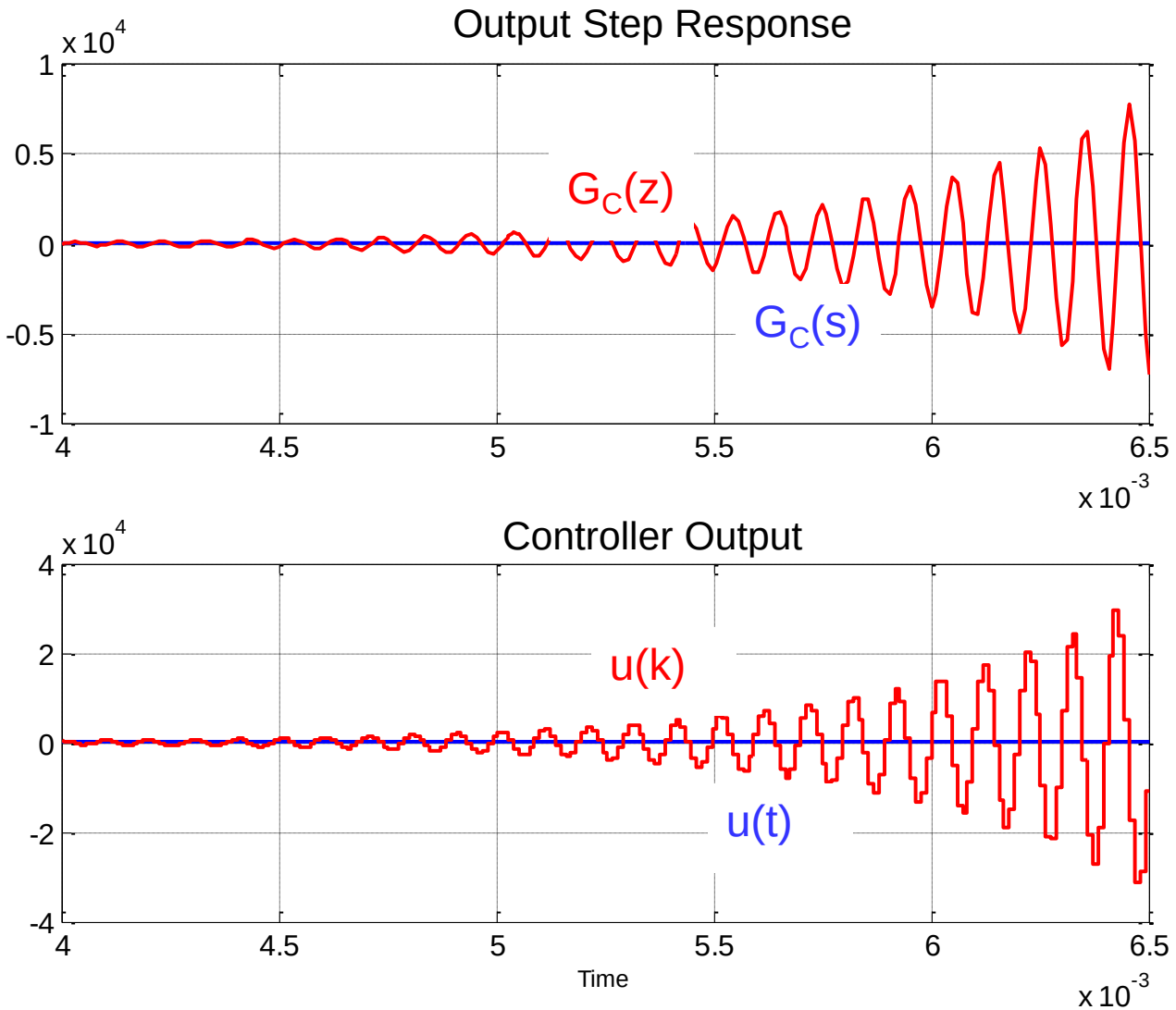


Digital Control Loop Design – Designing the Control Loop

- $G_C(s)$ replaced by $G_C(z)$, $T_s = 12\ \mu s$

$$G_C(z) = \frac{4.972 + 0.43z^{-1} - 4.542z^{-2}}{1 - 0.4193z^{-1} - 0.5807z^{-2}}$$

- Increasing sampling time to 12 μs the digital control loop becomes unstable

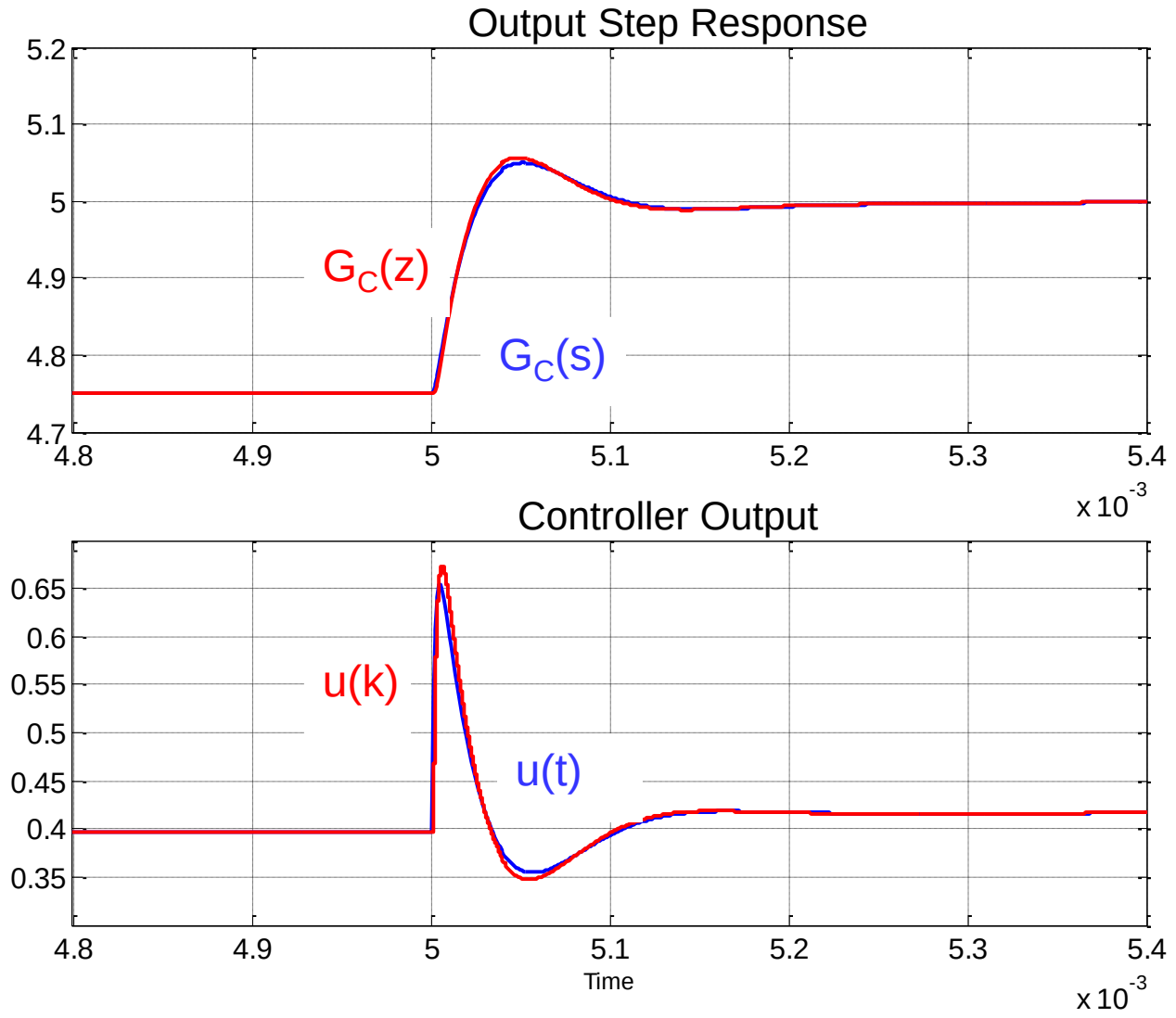


Digital Control Loop Design – Designing the Control Loop

- $G_C(s)$ replaced by $G_C(z)$, $T_s = 1\ \mu s$

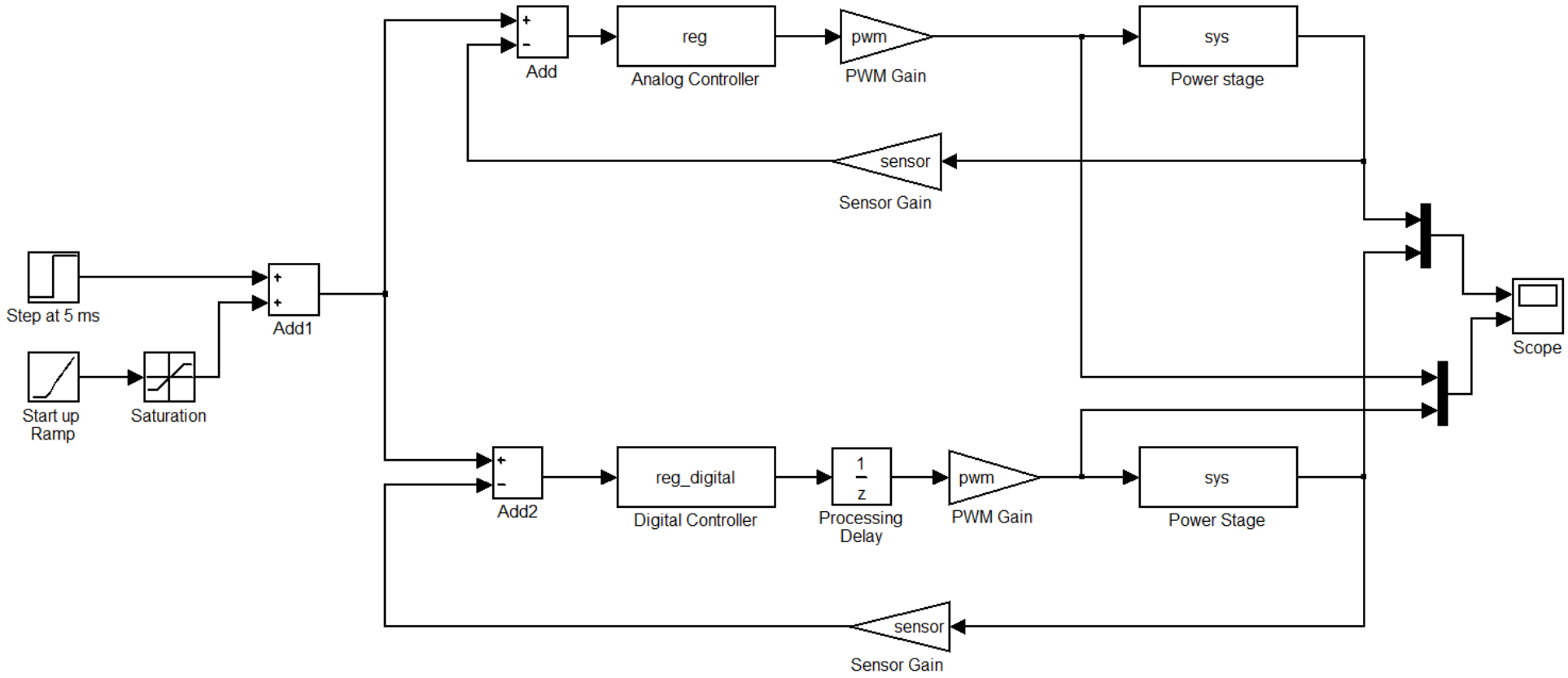
$$G_C(z) = \frac{1.444 + 0.01084z^{-1} - 1.433z^{-2}}{1 - 1.522z^{-1} - 0.5219z^{-2}}$$

- Increasing sample rate by 5x brings performance very close analog controller



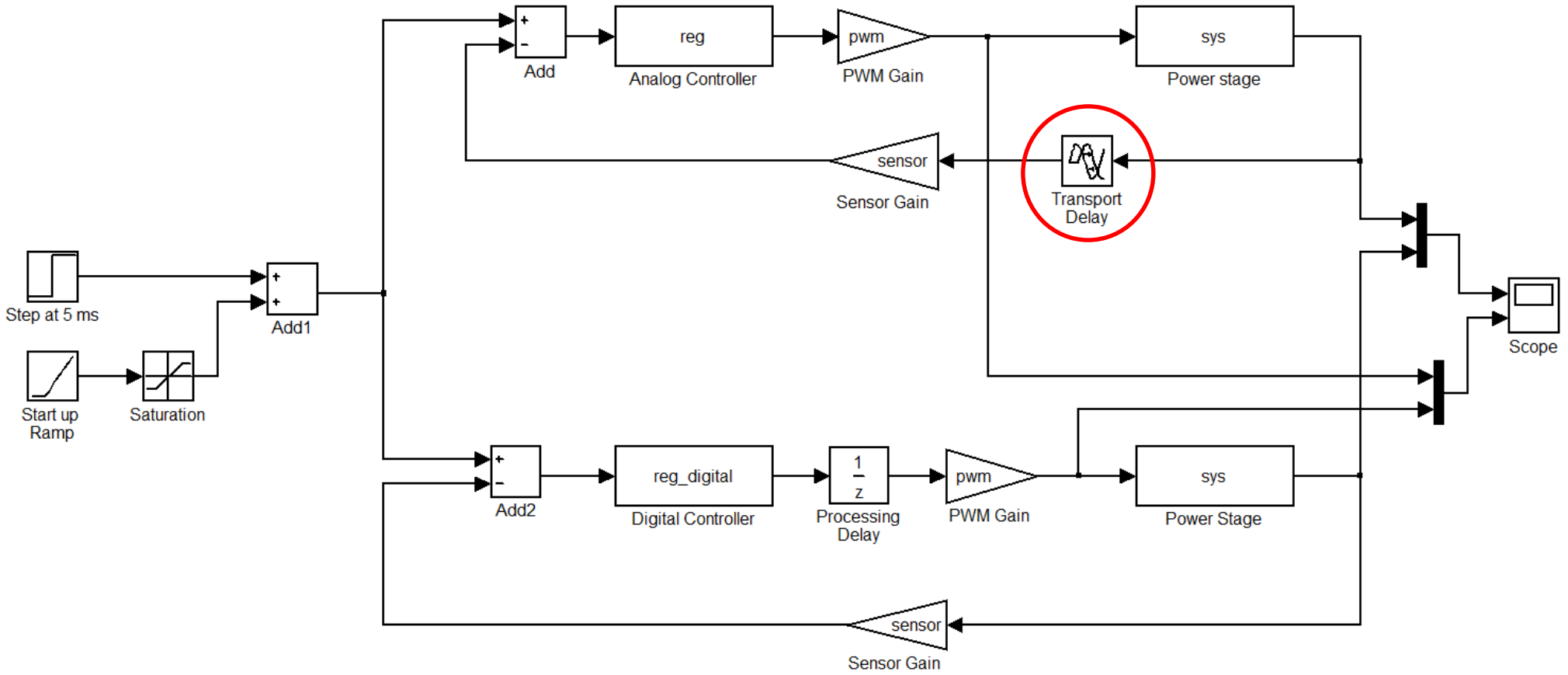
Digital Control Loop Design – Designing the Control Loop

- $G_C(s)$ replaced by $G_C(z)$ – MATLAB Model



Digital Control Loop Design – Designing the Control Loop

- $G_C(s)$ replaced by $G_C(z)$ – MATLAB Model with additional delay in analog control loop

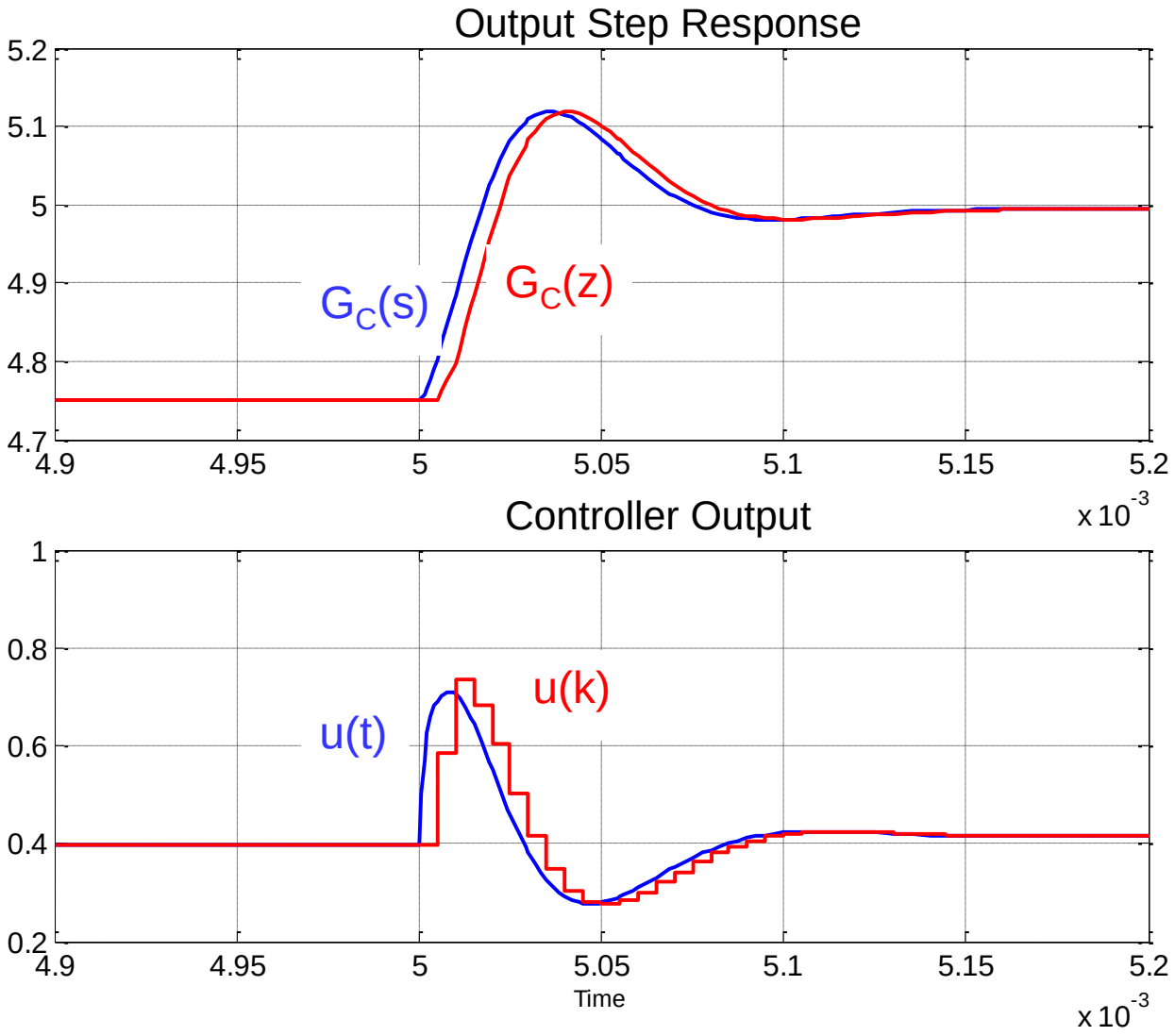


Digital Control Loop Design – Designing the Control Loop

- $G_C(s)$ replaced by $G_C(z)$, $T_s = 5\ \mu s$
- Added additional delay added into control loop

$$G_C(z) = \frac{3.747 + 0.1385z^{-1} - 3.608z^{-2}}{1 - 0.778z^{-1} - 0.222z^{-2}}$$

- Adding transport delay into analog loop doesn't bring expected response, but the digital control loop matches analog control loop performance



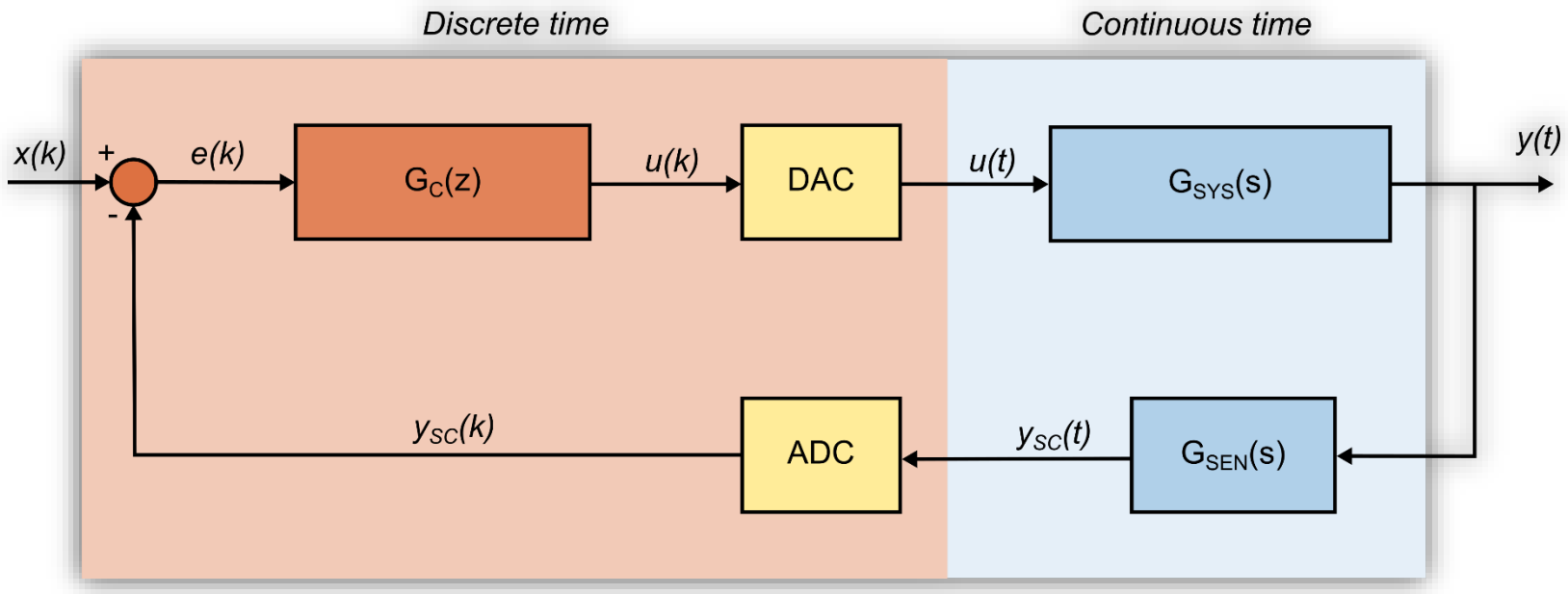
Digital Control Loop Design – Designing the Control Loop

- Control loop design by analog approach – Summary
  - Analog approach doesn’t respect delays caused by digitalization of control loop
  - If sampling of the system is high enough, it gives good performance similar to analog control loop

Where do the delays come from?
How long are the delays?

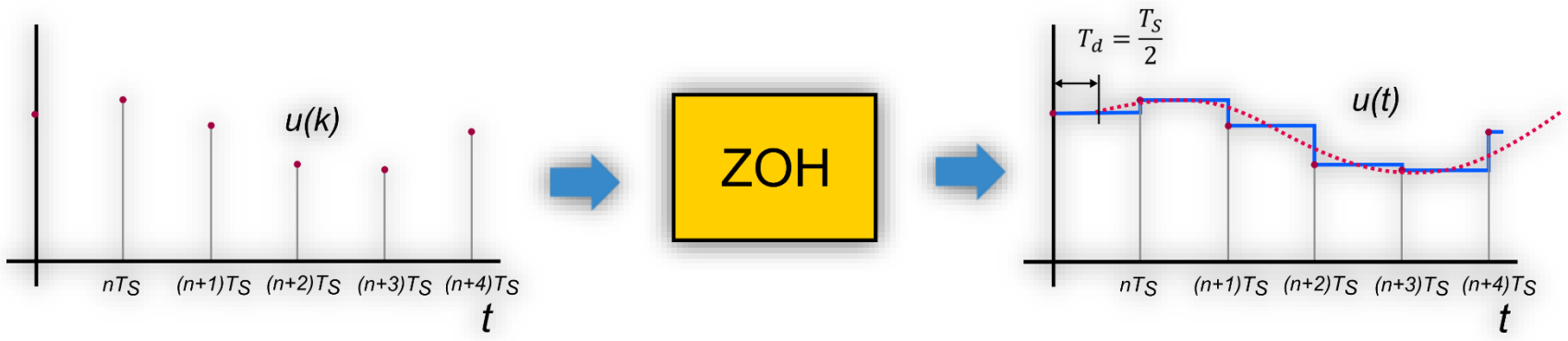
Digital Control Loop Design – Signal Reconstruction

- Signal Reconstruction from digital into continuous domain - u(k) → u(t)



Digital Control Loop Design – Signal Reconstruction

- Digital to Analog Converter - DAC
  - The DAC keeps constant output over whole period
  - The DAC is typical representative of so called zero-order hold circuit (ZOH)
  - The ZOH circuit introduces transport delay $T_S/2$

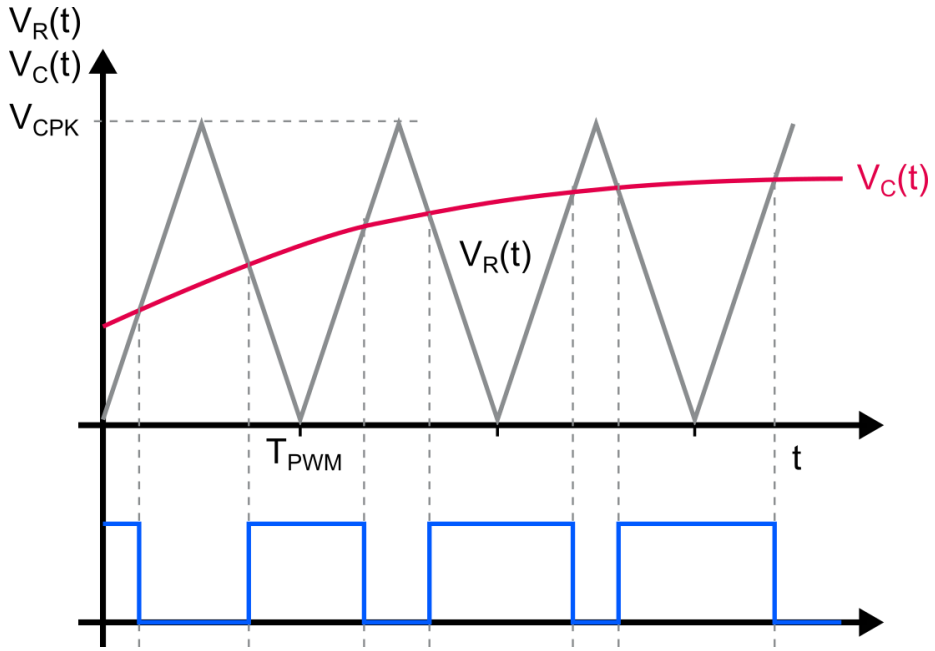


Digital Control Loop Design – Signal Reconstruction

- PWM Modulator
  - The SMPS uses PWM modulator for signal reconstruction
  - Since the SPMS contains energy storage elements (inductors, capacitors) the voltage/current is averaged and corresponds duty cycle of PWM modulator

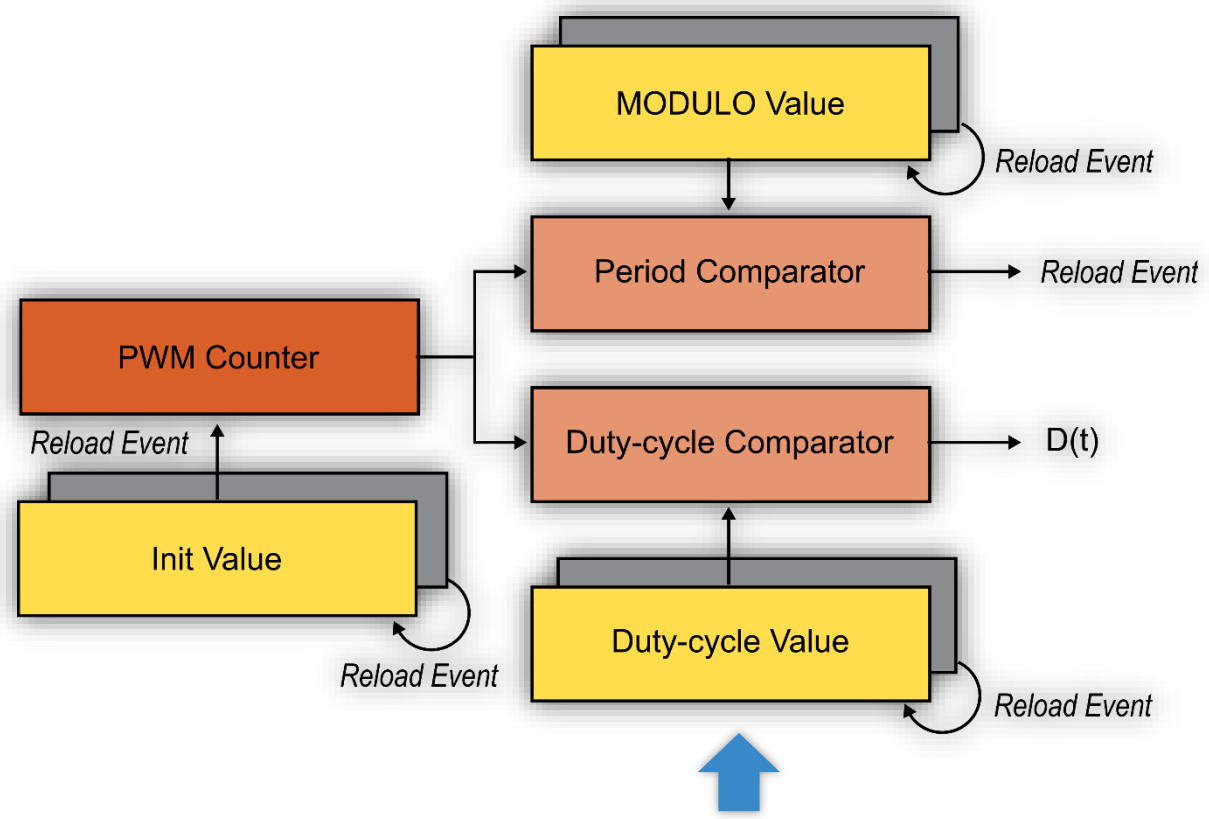
Digital Control Loop Design – Signal Reconstruction

- Analog PWM Modulator
  - The output of the controller $v_C(t)$ is continuously compared with triangle reference voltage $v_R(t)$. The PWM output is 1 when $v_C(t) > v_R(t)$, otherwise the output is zero
  - The analog PWM modulator introduces negligible delay in control loop



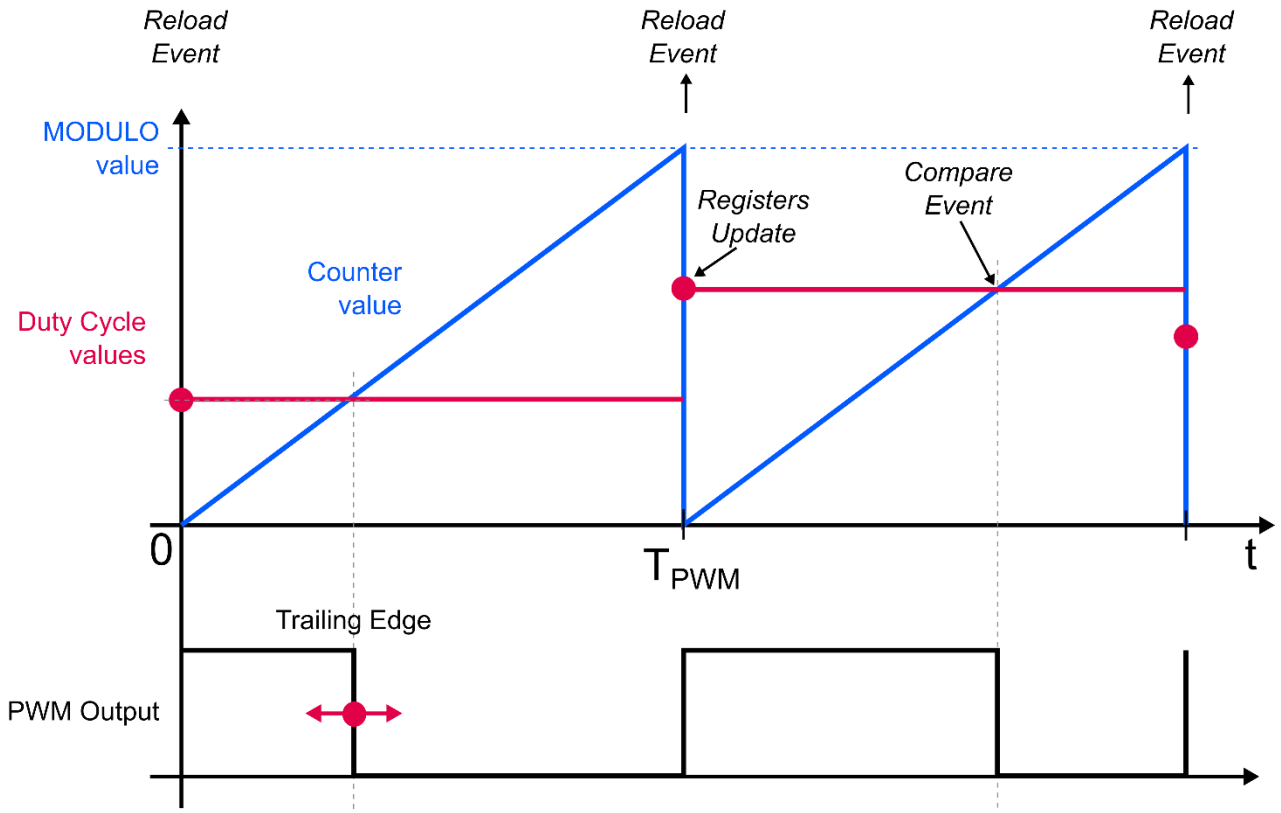
Digital Control Loop Design – Signal Reconstruction

- Digital PWM Modulator
  - The PWM frequency is defined by MODULO value, the duty cycle by Duty-cycle value. Both values are compared by digital comparators with PWM counter. The duty-cycle comparator generates PWM output and period comparator resets PWM counter and reloads all PWM related register from their buffers in order to generate consistent PWM output



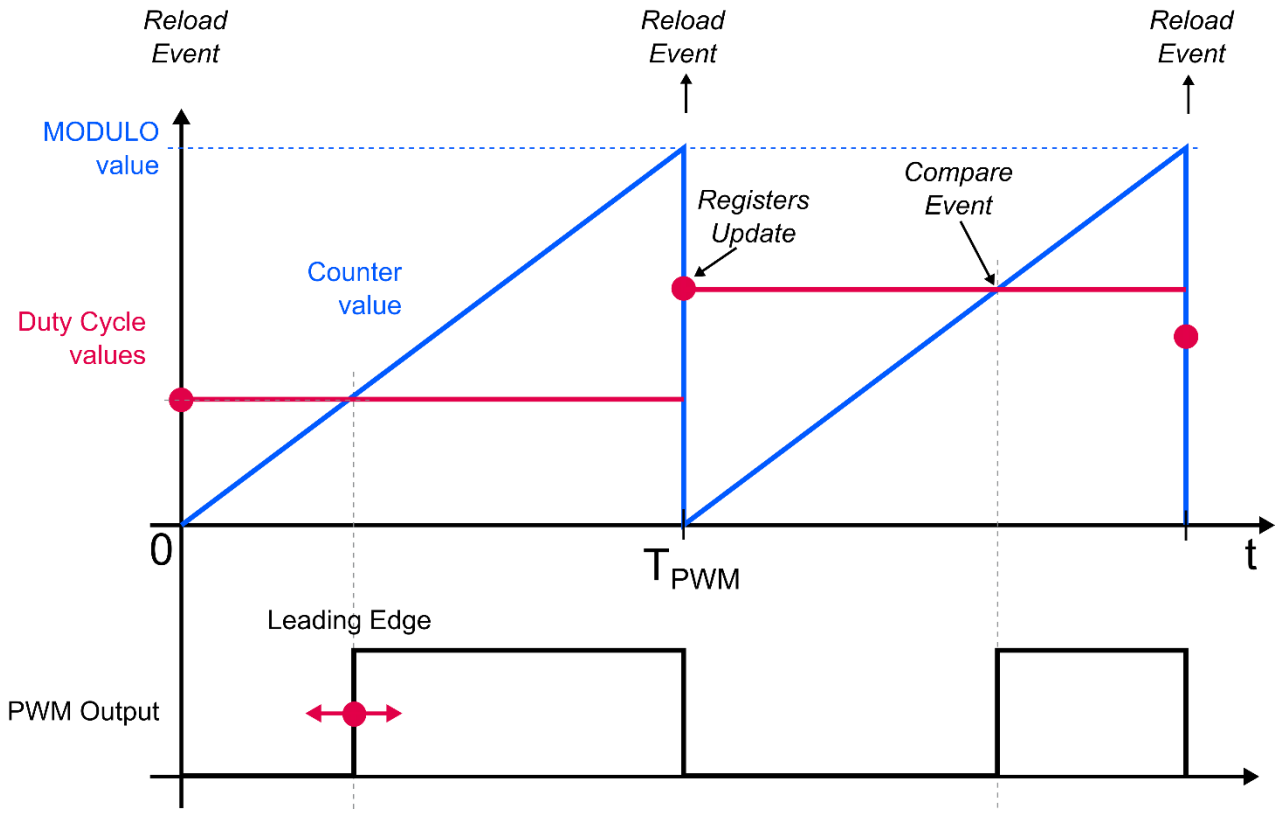
Digital Control Loop Design – Signal Reconstruction

- Digital PWM Modulator
  - Trailing edge modulation
    - The PWM pulse starts at the beginning of the period. The falling edge is moving according to duty cycle value



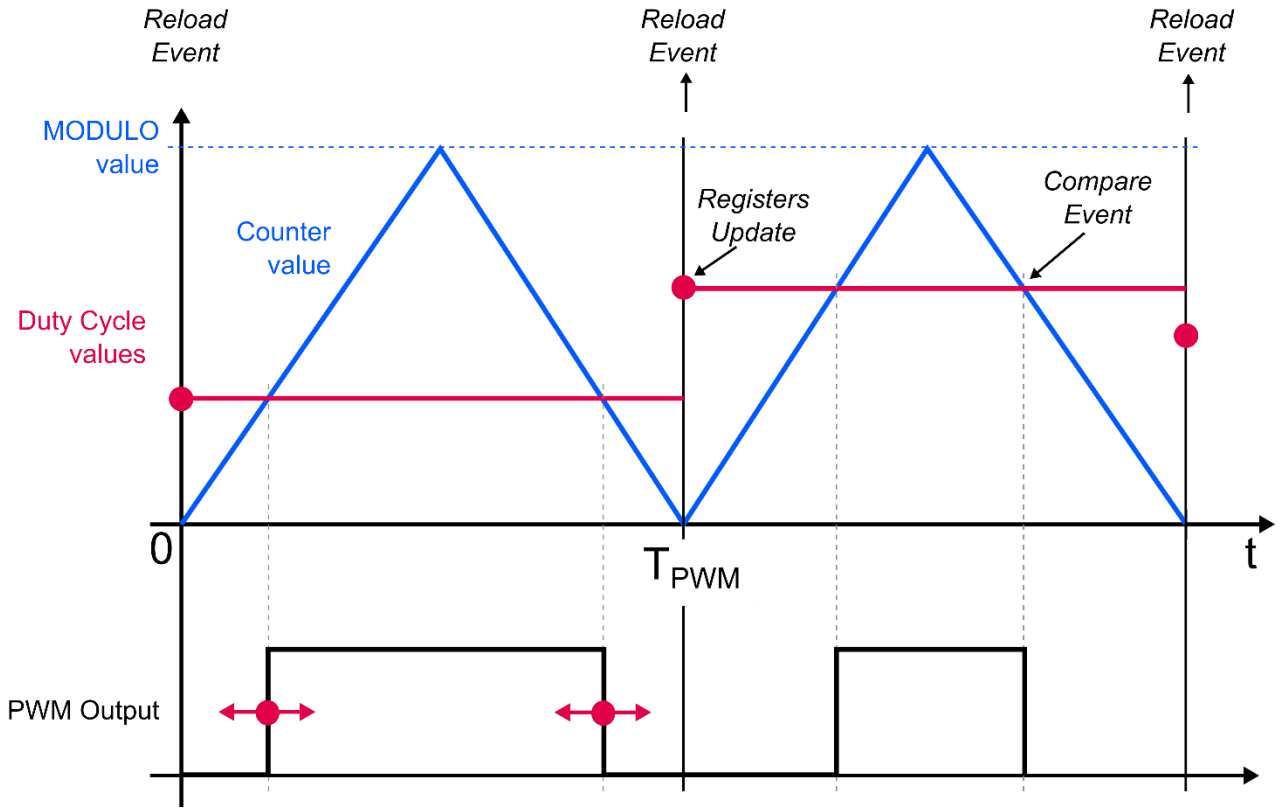
Digital Control Loop Design – Signal Reconstruction

- Digital PWM Modulator
  - Leading edge modulation
    - The rising edge is moving according to duty cycle value. The PWM pulse ends at the end of the period.



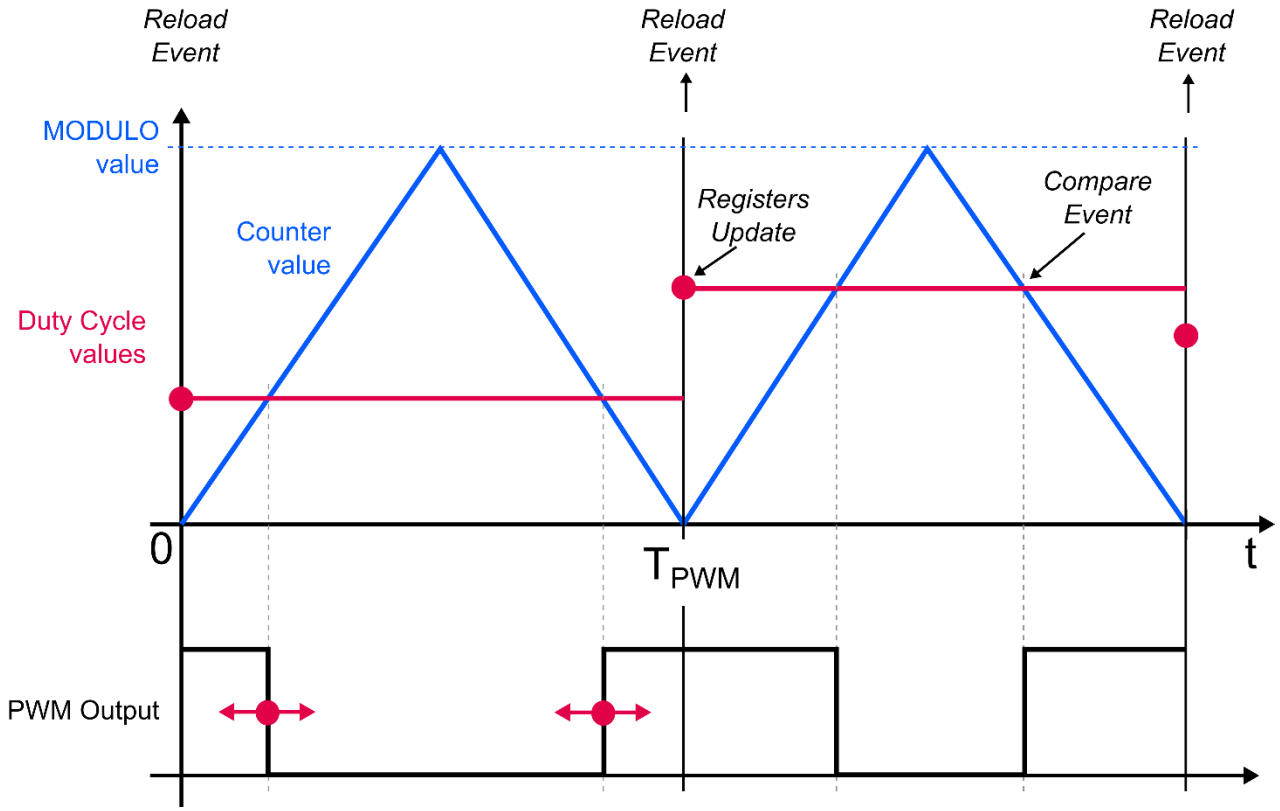
Digital Control Loop Design – Signal Reconstruction

- Digital PWM Modulator
  - Center-aligned PWM
    (center in the middle of the pulse)
    The pulse is symmetrical to the middle of the pulse. Both edges are moving according to the duty cycle value



Digital Control Loop Design – Signal Reconstruction

- Digital PWM Modulator
  - Center-aligned PWM
    (center at the beginning of the pulse)
    The pulse is symmetrical to the beginning of the pulse. Both edges are moving according to the duty cycle value

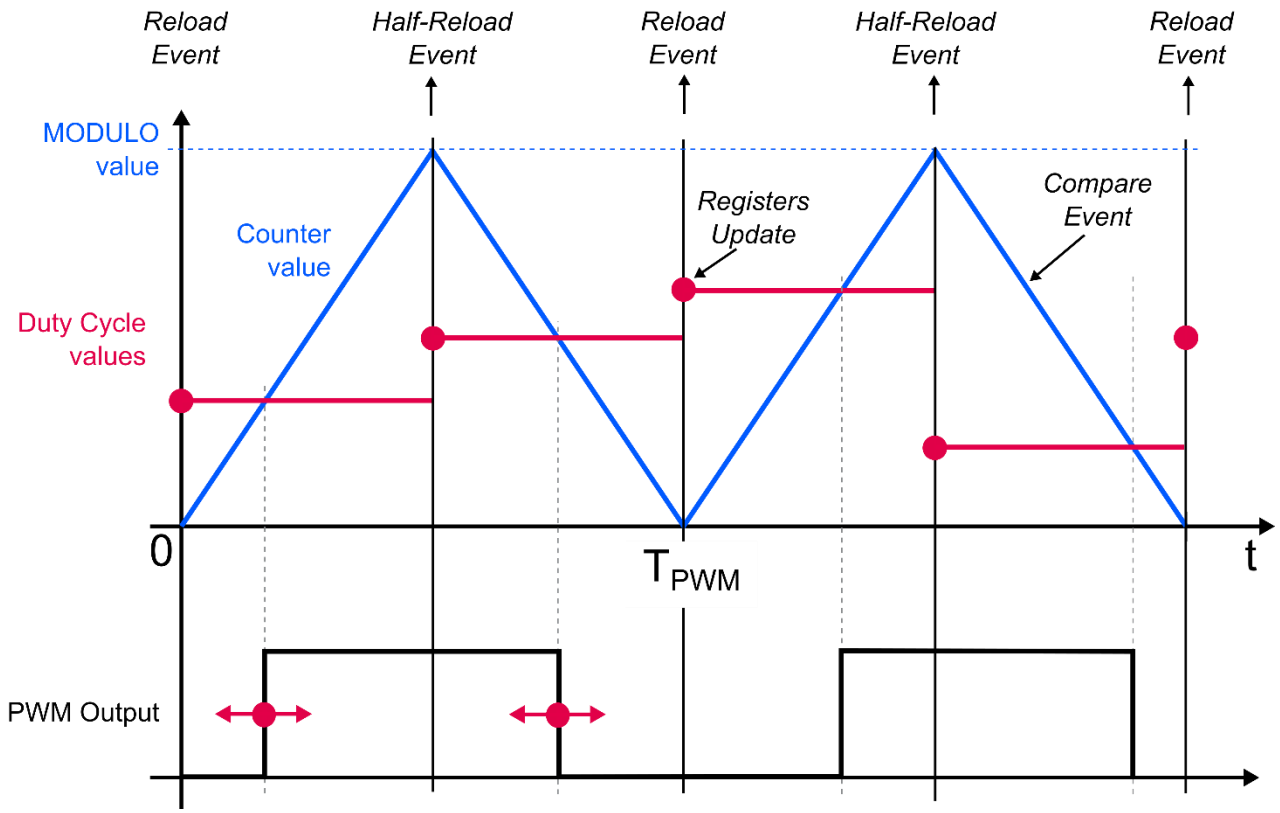


Digital Control Loop Design – Signal Reconstruction

- Digital PWM Modulator
  - Double reload PWM

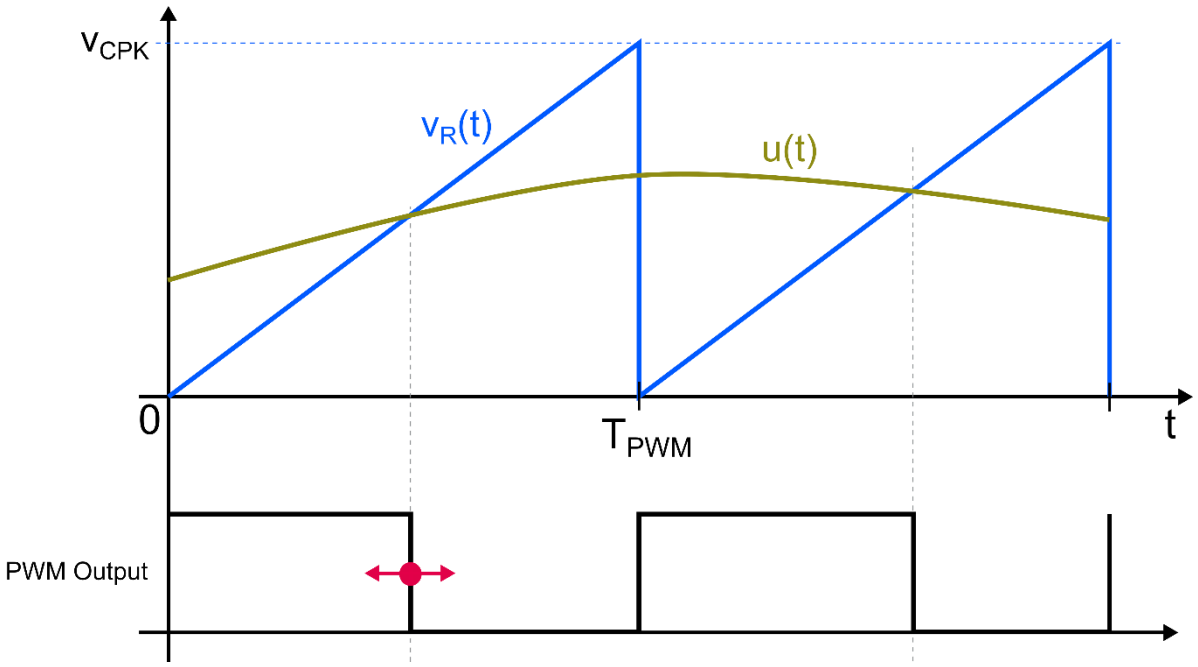
    There is additional reload event in the middle of the PWM period, where duty cycle is updated

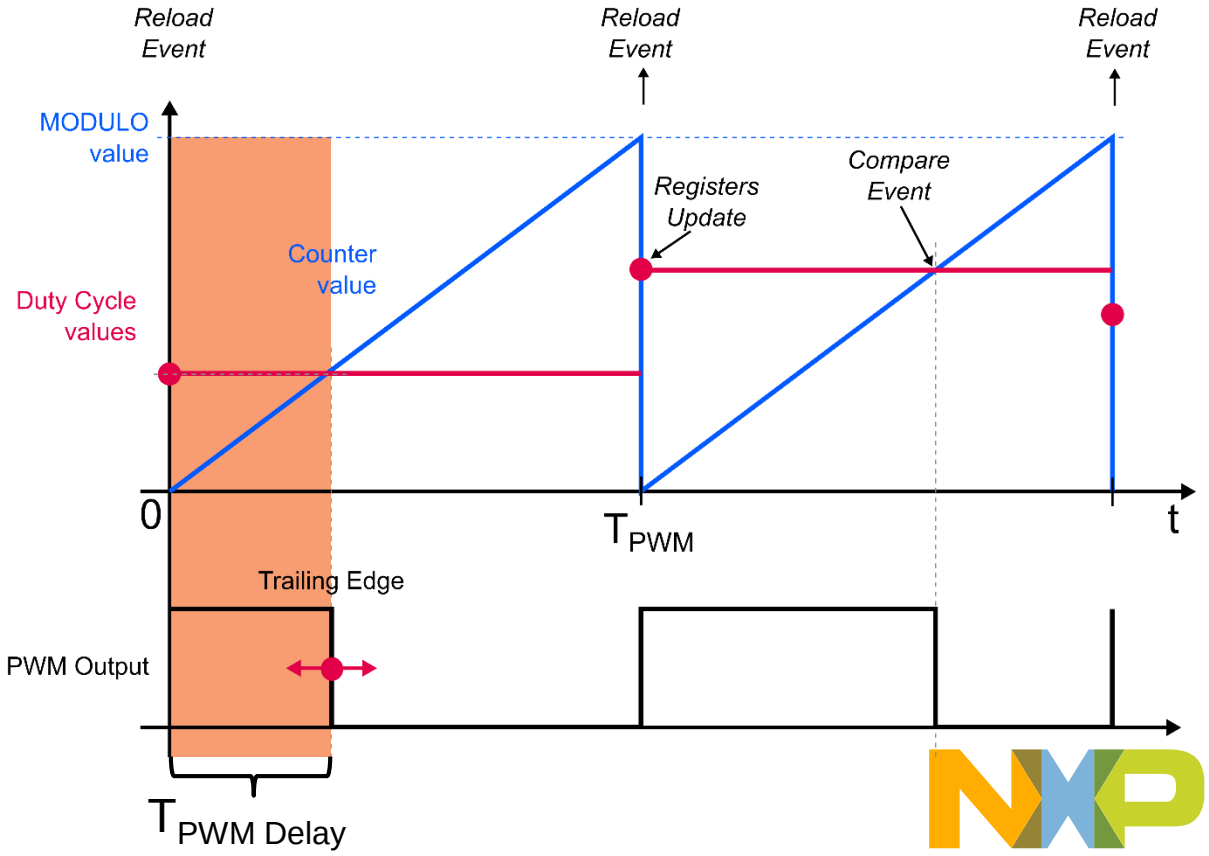
    The pulse is asymmetrical to the middle of the pulse. Both edges are moving according to the duty cycle values



Digital Control Loop Design – Signal Reconstruction

- Analog versus Digital PWM Modulator
  - In analog PWM modulator the falling edge is generated by actual controller output u(t) while in digital PWM the falling edge is generated by digital controller output u(k), which was set at the beginning of PWM period





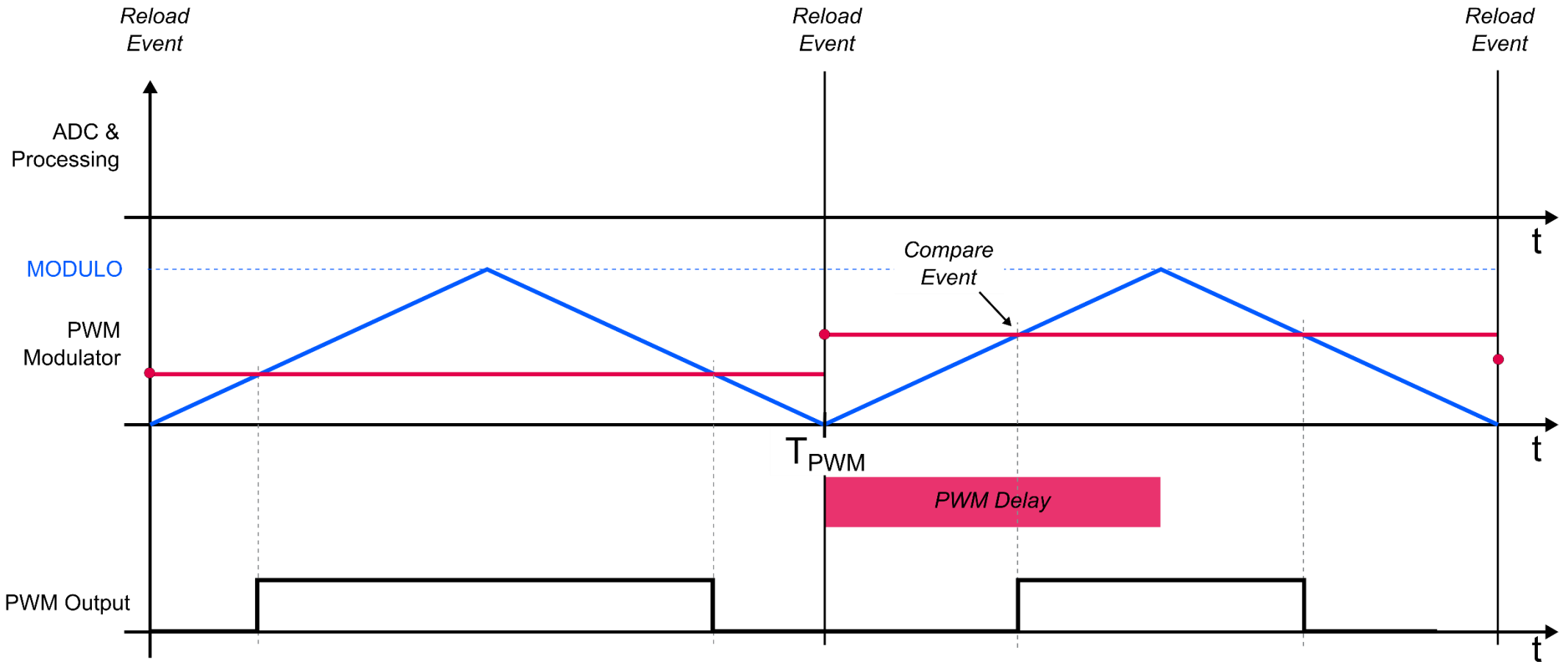
Digital Control Loop Design – Signal Reconstruction

- Analog versus Digital PWM Modulator
  - Digital PWM modulator introduces time delay against to analog PWM modulator
  - The time delay differs with PWM modulation type – see table below

PWM Modulation	Delay
Trailing edge PWM	DT_{PWM}
Leading edge PWM	$(1-D)T_{PWM}$
Center aligned PWM	$\frac{T_{PWM}}{2}$
Double reload PWM	$\frac{T_{PWM}}{4}$

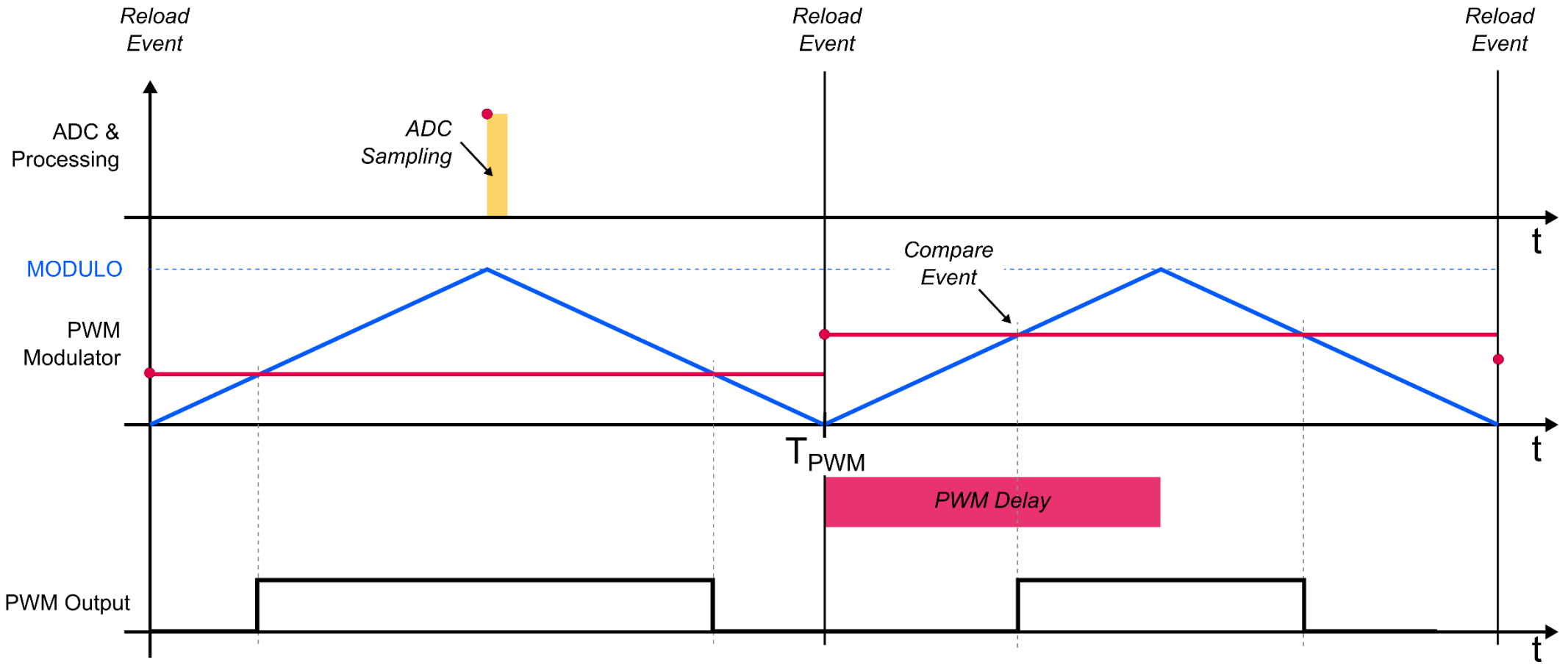
Digital Control Loop Design – Processing Delay

- Is the PWM modulator delay the only delay in the digital control loop?



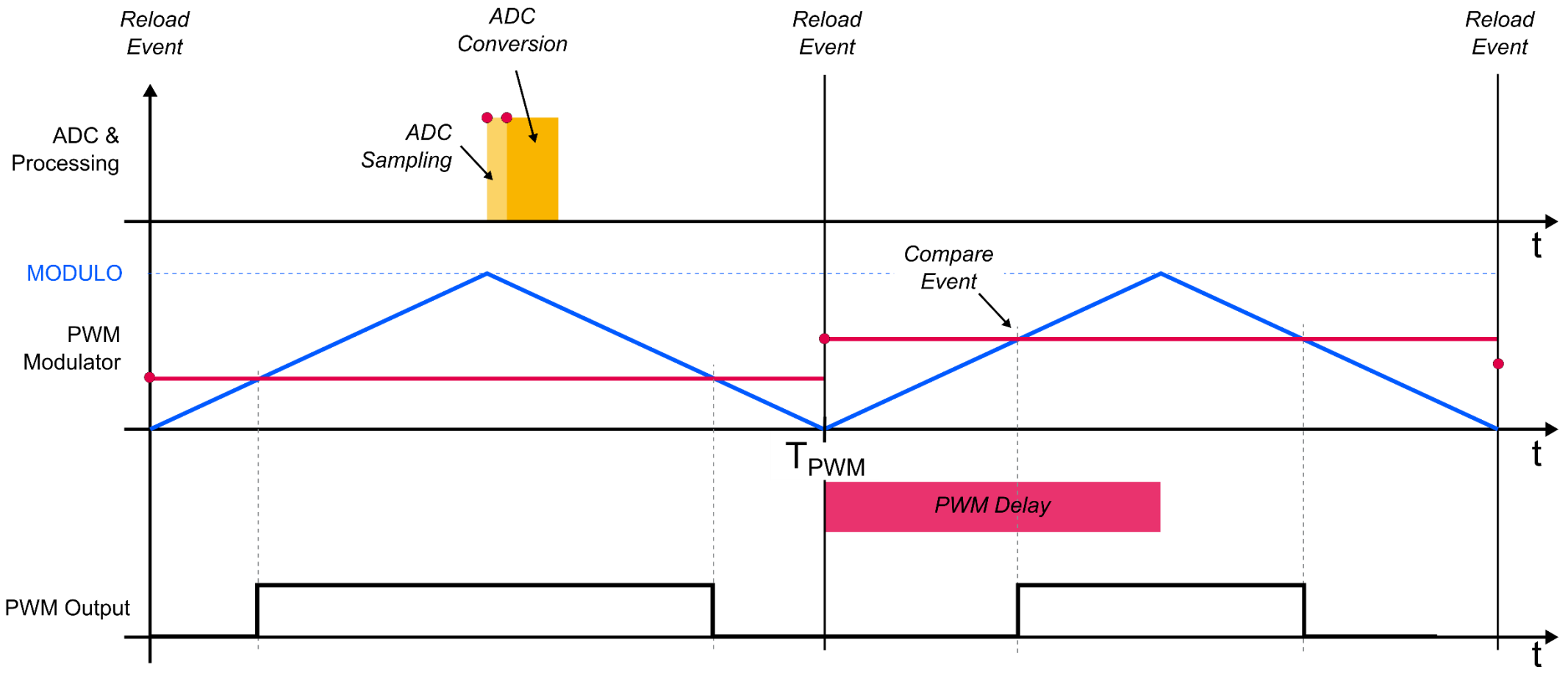
Digital Control Loop Design – Processing Delay

- Is the PWM modulator delay the only delay in the digital control loop?



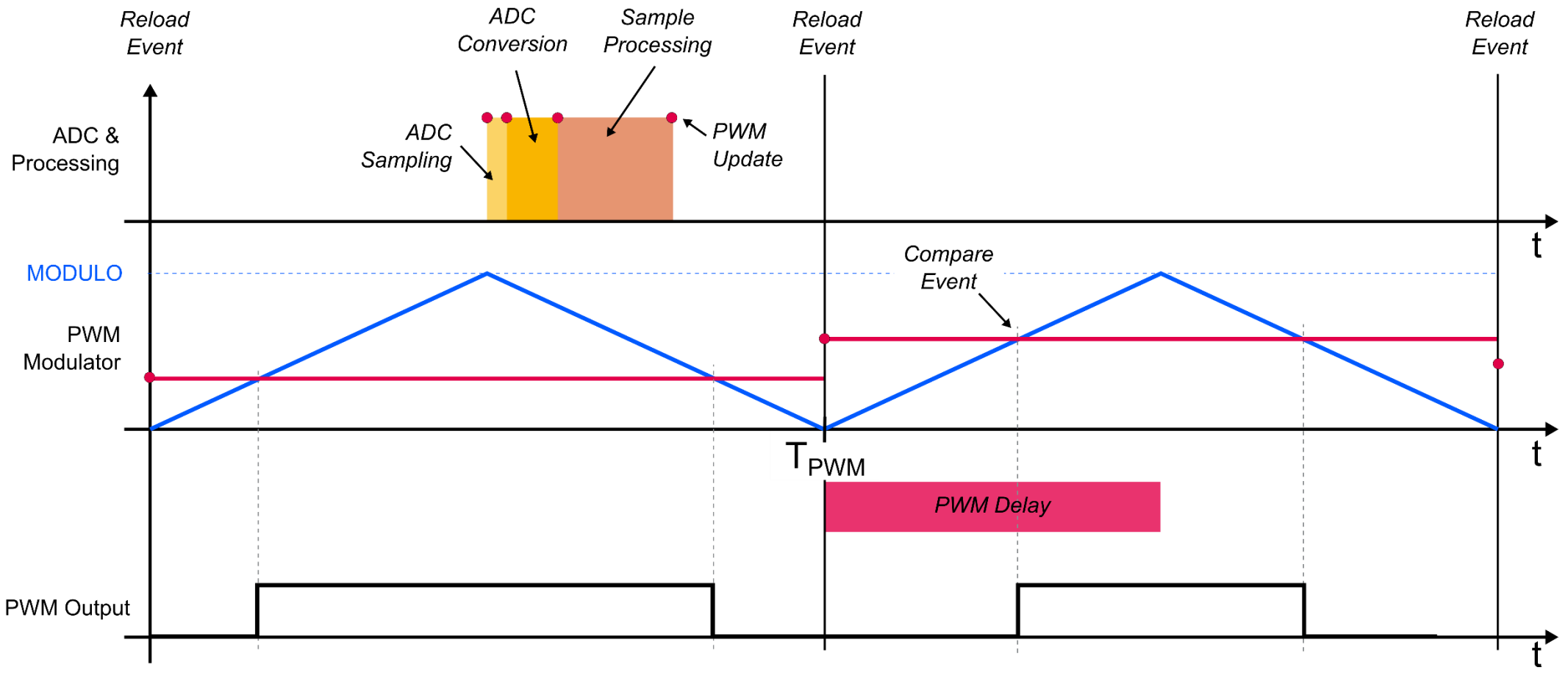
Digital Control Loop Design – Processing Delay

- Is the PWM modulator delay the only delay in the digital control loop?



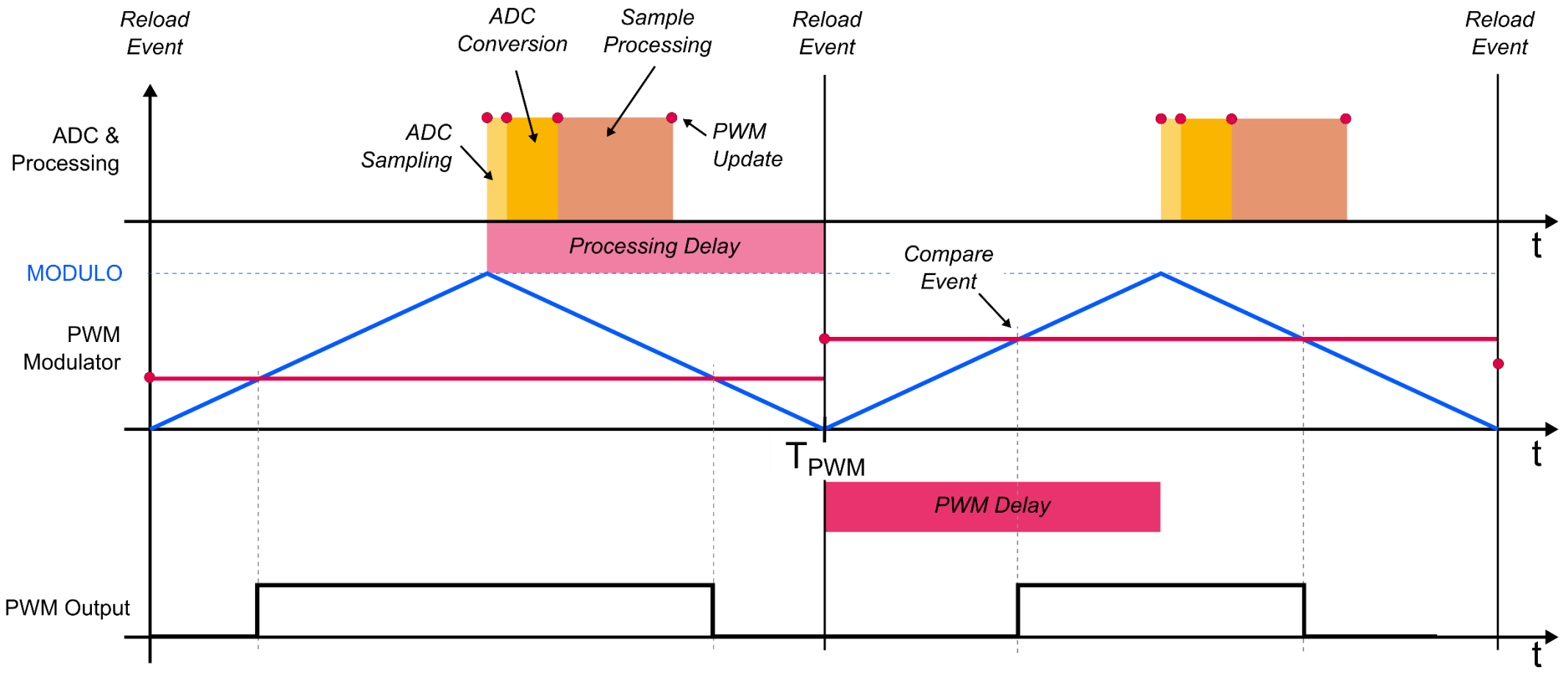
Digital Control Loop Design – Processing Delay

- Is the PWM modulator delay the only delay in the digital control loop?



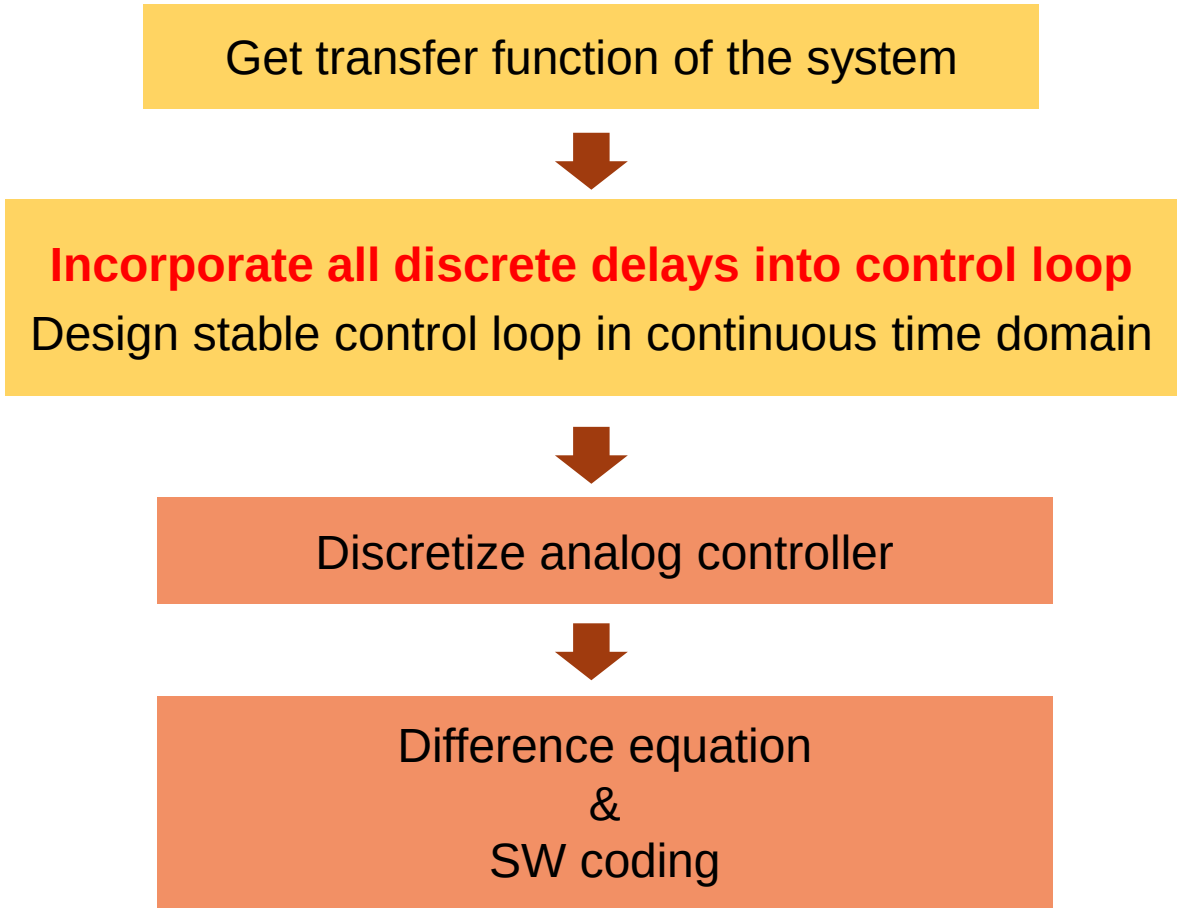
Digital Control Loop Design – Processing Delay

- Is the PWM modulator delay the only delay in the digital control loop?



Digital Control Loop Design – Designing the Control Loop

Modified Design by Emulation (Analog Approach)



Digital Control Loop Design – Designing the Control Loop

- How to integrate transport delay into control loop?
  - The transport delay is given by following irrational function

$$G_{TD}(s) = e^{-sT_{TD}}$$

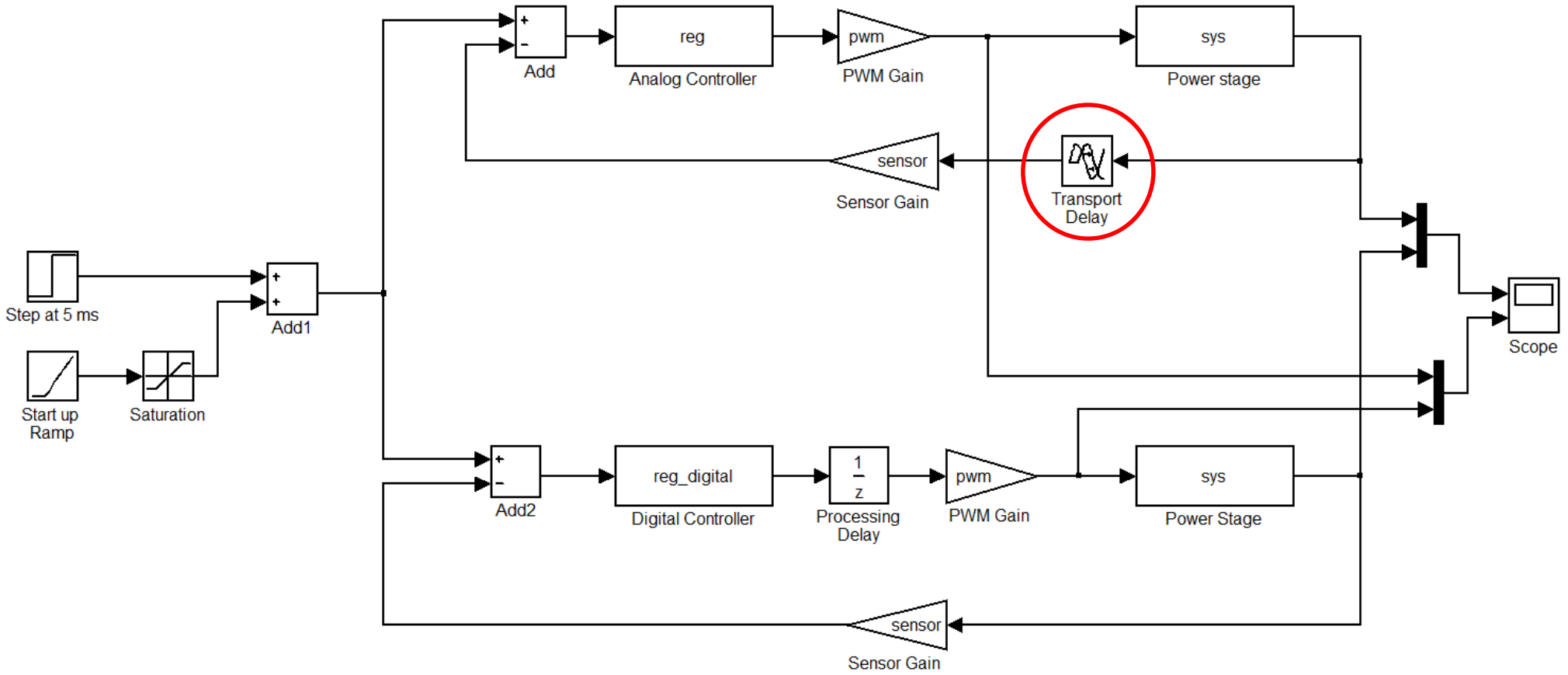
  - This function can be approximated by Padé approximation

$$G_{TD}(s) = e^{-sT_{TD}} \approx \frac{1-s\frac{T_{TD}}{2}}{1+s\frac{T_{TD}}{2}}$$

  - Where T_{TD} is total delay in digital control loop (processing + PWM modulator delays)

Digital Control Loop Design – Designing the Control Loop

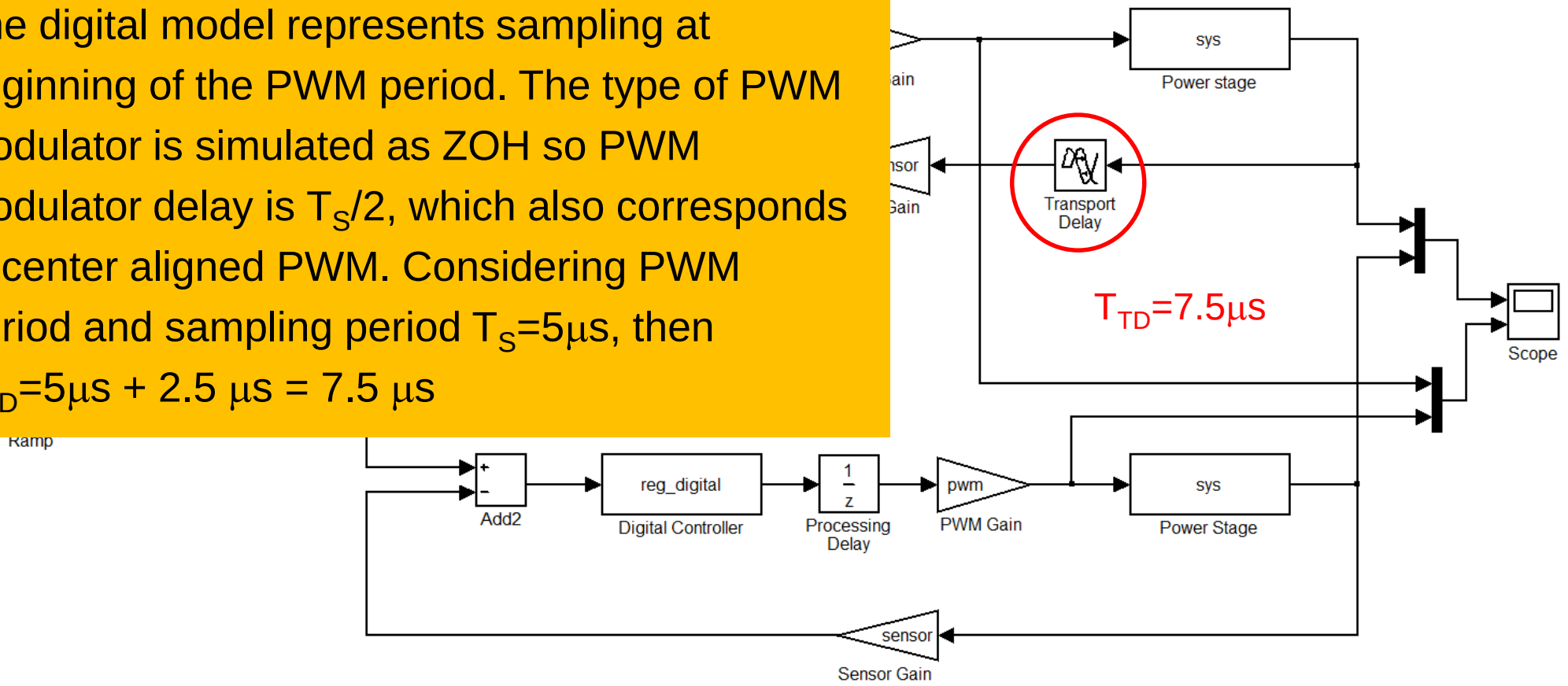
- Now we can evaluate transport delay in control loop



Digital Control Loop Design – Designing the Control Loop

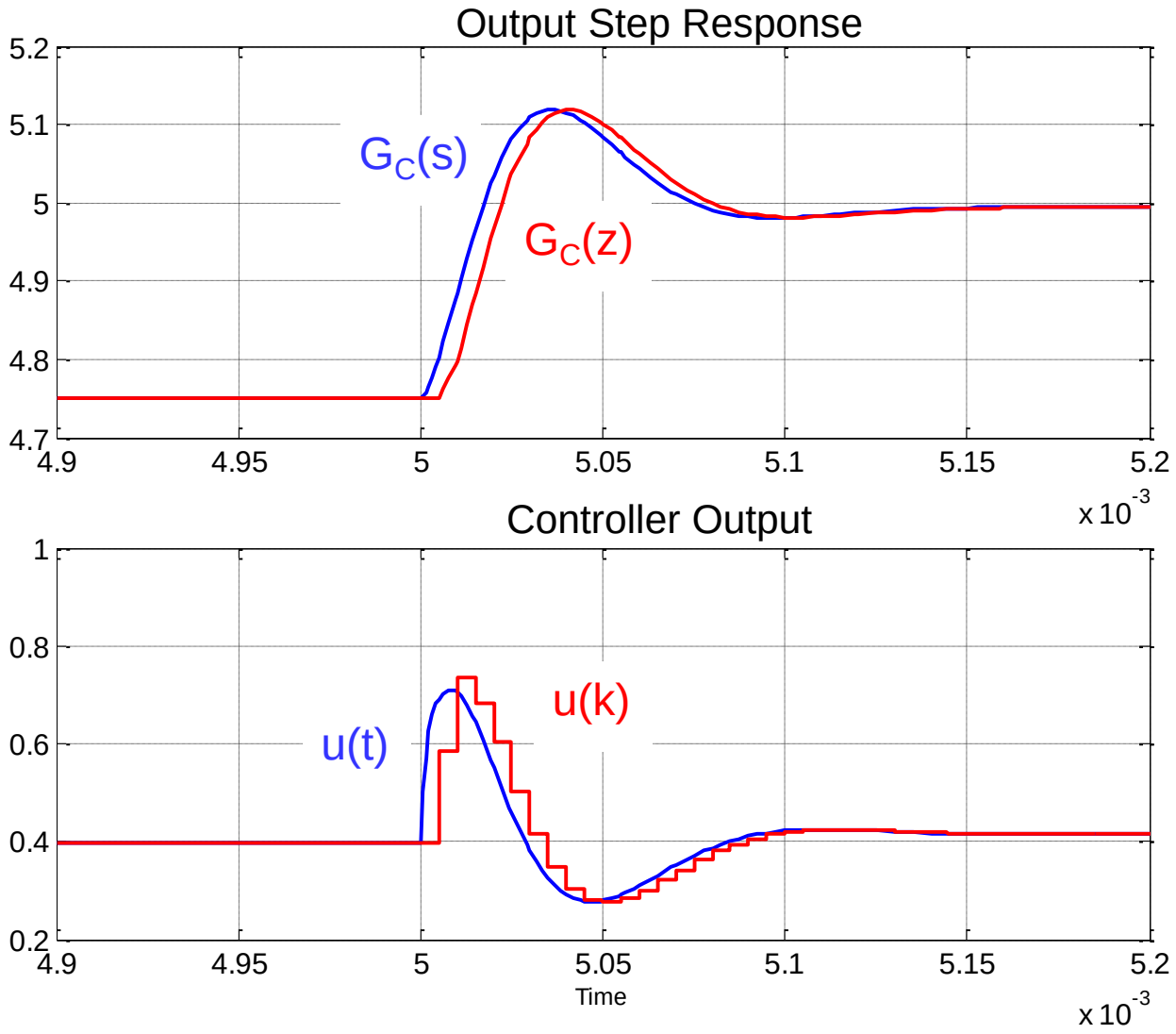
- Now we can evaluate transport delay in control loop

The digital model represents sampling at beginning of the PWM period. The type of PWM modulator is simulated as ZOH so PWM modulator delay is $T_S/2$, which also corresponds to center aligned PWM. Considering PWM period and sampling period T_S=5µs, then T_{TD}=5µs + 2.5 µs = 7.5 µs



Digital Control Loop Design – Designing the Control Loop

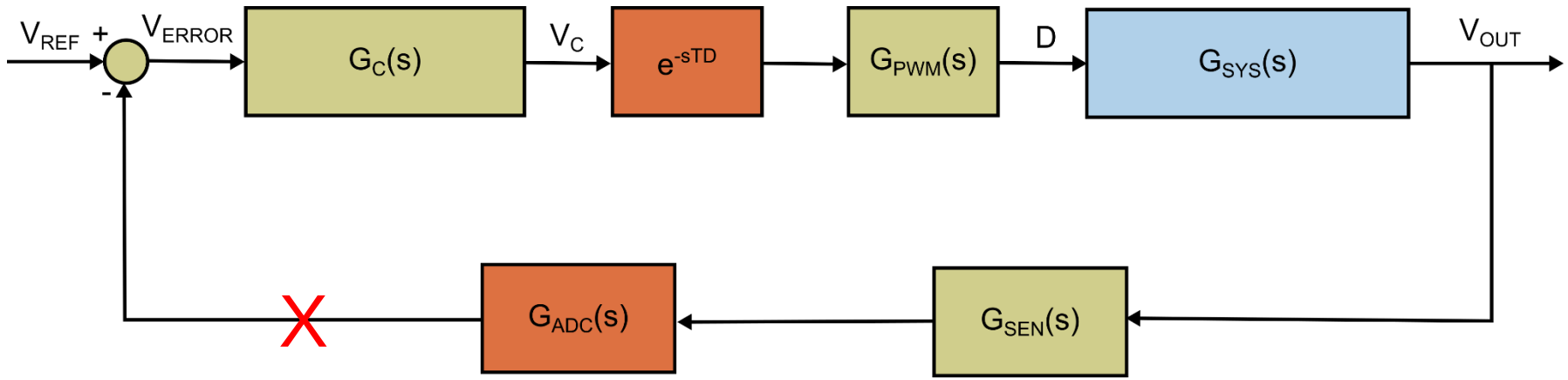
- Since we are able to identify delay caused by digitalization of the control loop we can repeat design of the controller using the same (analog) approach but respecting delays caused by digital implementation



Digital Control Loop Design - Buck Converter Example

- Buck converter open-control loop including transport delay

$$G_{OPEN}(s) = G_c(s) \cdot G_{TD}(s) \cdot G_{PWM}(s) \cdot G_{SYS}(s) \cdot G_{SEN}(s) \cdot G_{ADC}(s)$$



Digital Control Loop Design - Buck Converter Example

- Open loop transfer function

  Lets $G_{PWM}(s) = 1$, $G_{ADC}(s) = 1$, $T_{TD} = 7.5\mu s$ and $G_{SE}(s) = 0.2$ then

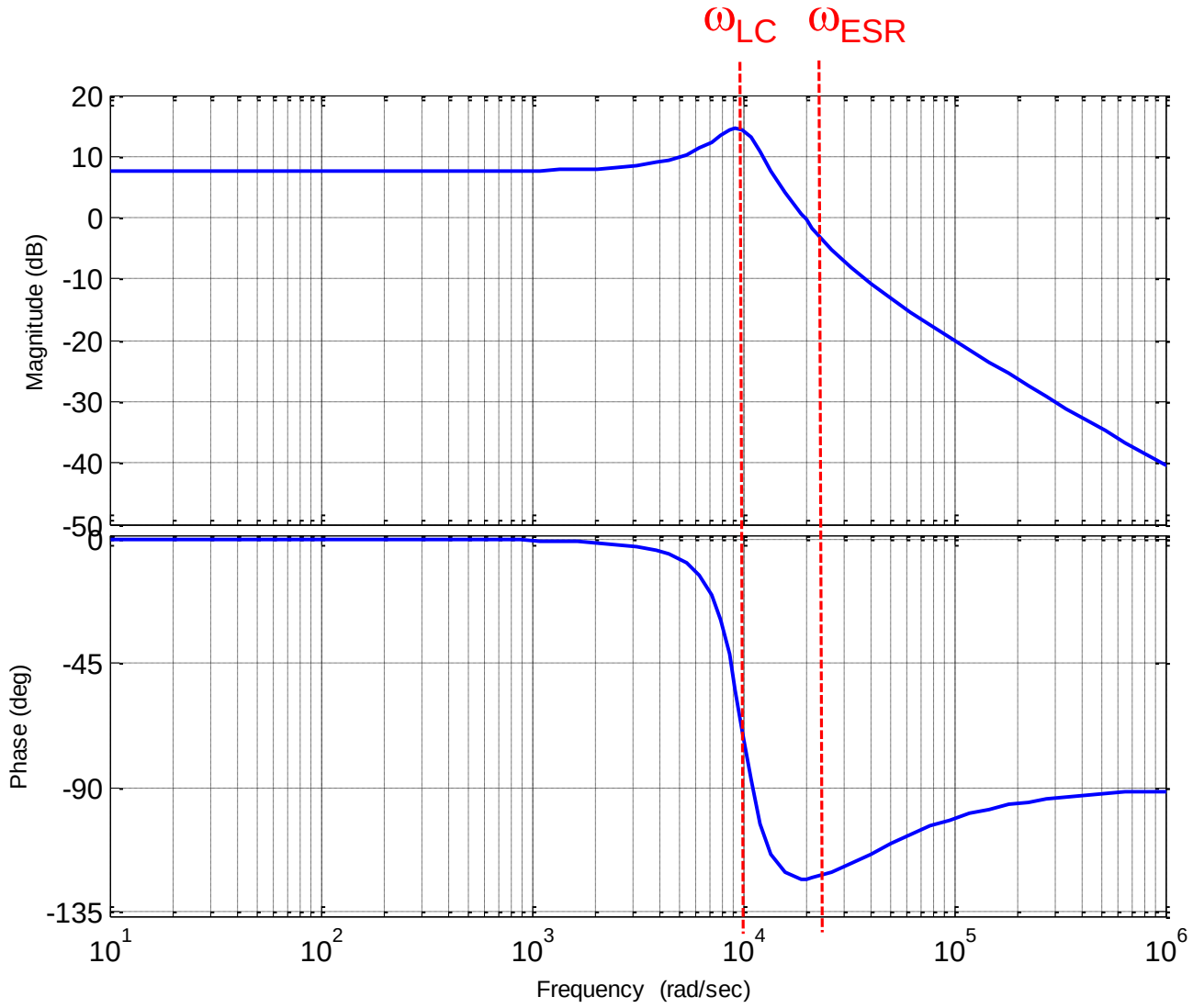
$$G_{OPEN}(s) = \frac{v_{out}(s)}{v_{ref}(s)} = \underbrace{\frac{K(1-\frac{s}{z_1})}{s\,(1-\frac{s}{p_2})}}_{G_C(s)} \underbrace{\frac{1-s\frac{7.5\cdot10^{-6}}{2}}{1+s\frac{7.5\cdot10^{-6}}{2}}}_{G_{TD}(s)} \underbrace{1}_{G_{PWM}(s)} \underbrace{\frac{1}{1}}_{G_{ADC}(s)} \underbrace{\frac{1}{5}}_{G_{SEN}(s)} \underbrace{\frac{37037.037(s+1.667\cdot10^4)}{s^2+4086s+5.144\cdot10^7}}_{G_{SYS}(s)}$$

Digital Control Loop Design - Buck Converter Example

- Controller Type II design - $G_C(s)$
  - Performance parameters of closed loop
    $\omega_c = 0.05 f_{sw} = 62.831 \cdot 10^3 \text{ rad/s}$
    $PM > 45°$
  - Controller Type II can be written as:

$$G_C = \frac{K(1 - \frac{s}{z_1})}{s\,(1 - \frac{s}{p_2})}$$

  - Place z_1 at $0.75\omega_{LC} = 7532 \text{ rad/s}$
  - Place p_2 at $\omega_2 = \frac{1}{2} 2\pi f_{sw} = 6.28 \cdot 10^5 \text{ rad/s}$
  - Adjust K to get desired loop bandwidth at ω_C



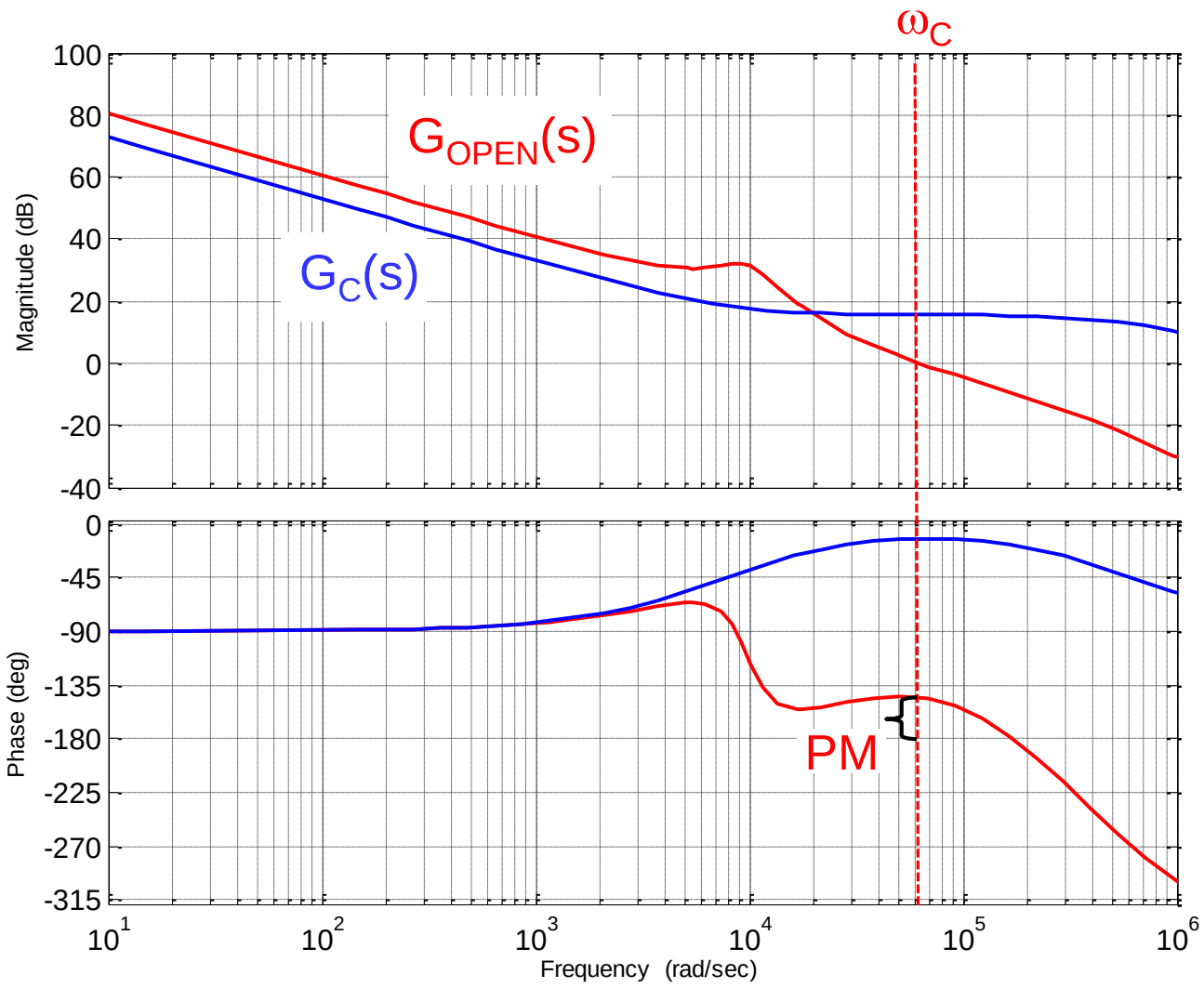
Digital Control Loop Design - Buck Converter Example

- Complete open loop $G_{OPEN}(s)$

$$G_{PWM}(s) \cdot G_{TD}(s) \cdot G_{SYS}(s) \cdot G_{SEN}(s) \cdot G_{ADC}(s) \cdot G_C(s)$$

$$G_C = \frac{6682585.71(s + 3402)}{s\,(s + 3.142 \cdot 10^5)}$$

- $\omega_C = 62.831 \cdot 10^3$
- $\boldsymbol{PM = 34.2° < 45°}$



Digital Control Loop Design - Buck Converter Example

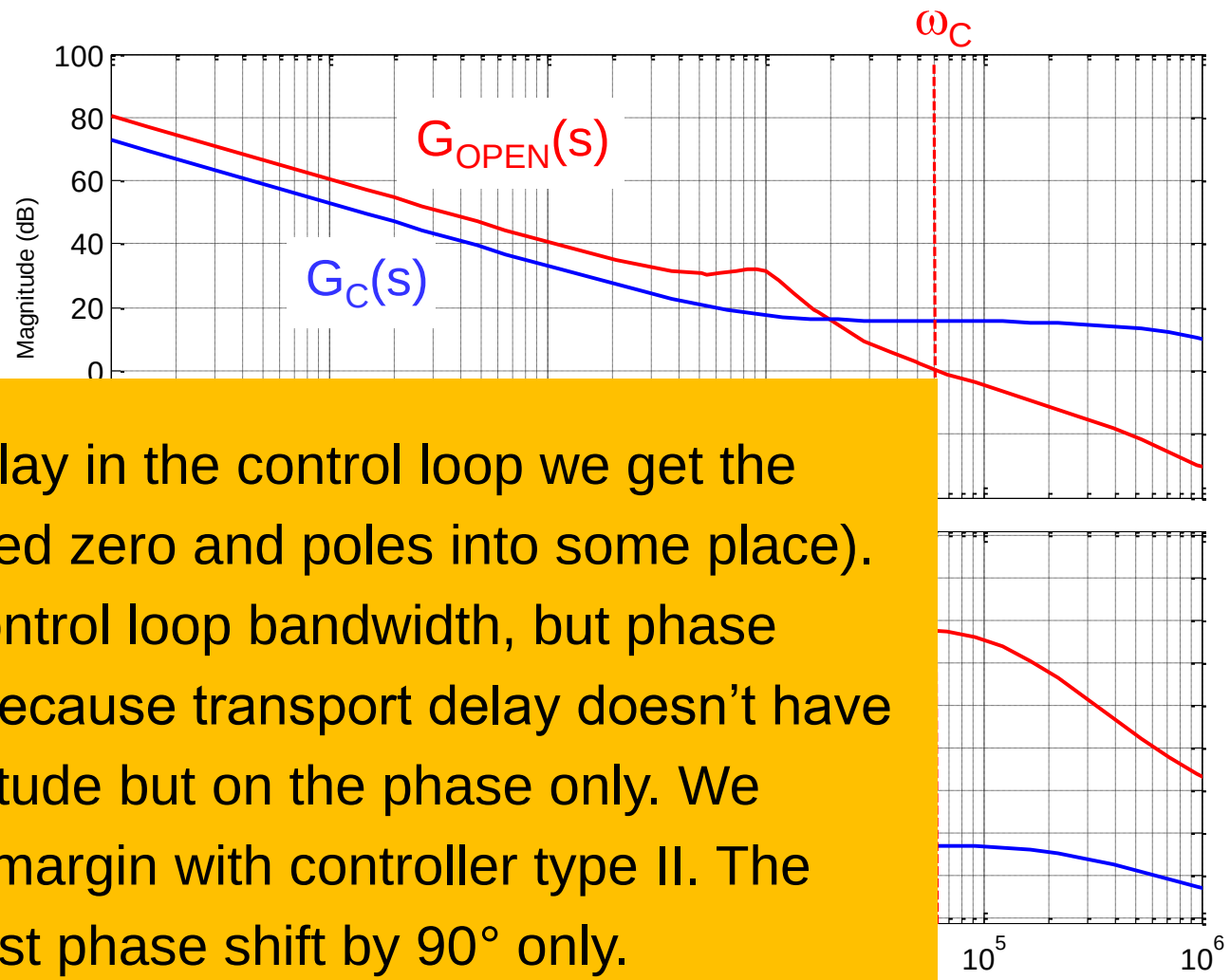
- Complete open loop $G_{OPEN}(s)$

$$G_{PWM}(s) \cdot G_{TD}(s) \cdot G_{SYS}(s) \cdot G_{SEN}(s)$$

$$\cdot G_{ADC}(s) \cdot G_C(s)$$

$$G_C = \frac{668258}{s\,(s +$$

- $\omega_C = 62.831$
- $\boldsymbol{PM = 34.2°}$



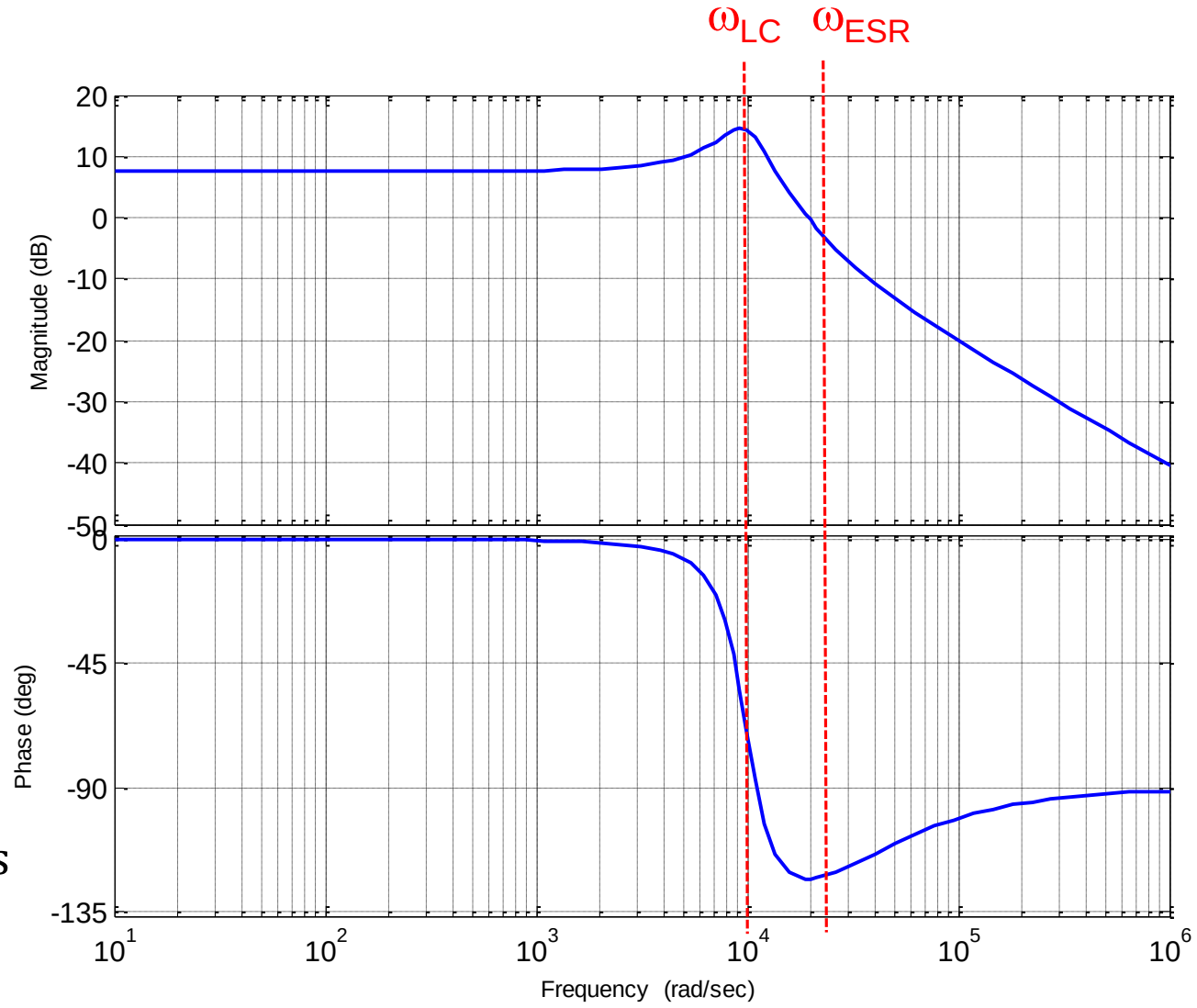
Considering transport delay in the control loop we get the same controller (we placed zero and poles into some place). We achieved required control loop bandwidth, but phase margin is 34.2°. This is because transport delay doesn't have any impact on the magnitude but on the phase only. We cannot get better phase margin with controller type II. The controller type II can boost phase shift by 90° only.

Digital Control Loop Design - Buck Converter Example

- Select Controller Type III
  - Performance parameters of closed loop
    $\omega_c = 0.05 f_{sw} = 62.831 \cdot 10^3 \text{ rad/s}$
    $PM > 45°$
  - Controller Type III:

$$G_C = \frac{K(1 - \frac{s}{z_1})(1 - \frac{s}{z_2})}{s\,(1 - \frac{s}{p_2})(1 - \frac{s}{p_3})}$$

  - Place z_1, z_1 at $0.75\omega_{LC} = 3402 \text{ rad/s}$
  - Place p_2 at $\omega_{ESR} = \frac{1}{R_{ESR}C} = 24510 \text{ rad/s}$
  - Place p_3 at $\omega_2 = \frac{1}{2} 2\pi f_{sw} = 6.28 \cdot 10^5 \text{ rad/s}$
  - Adjust K to get desired loop bandwidth at ω_C



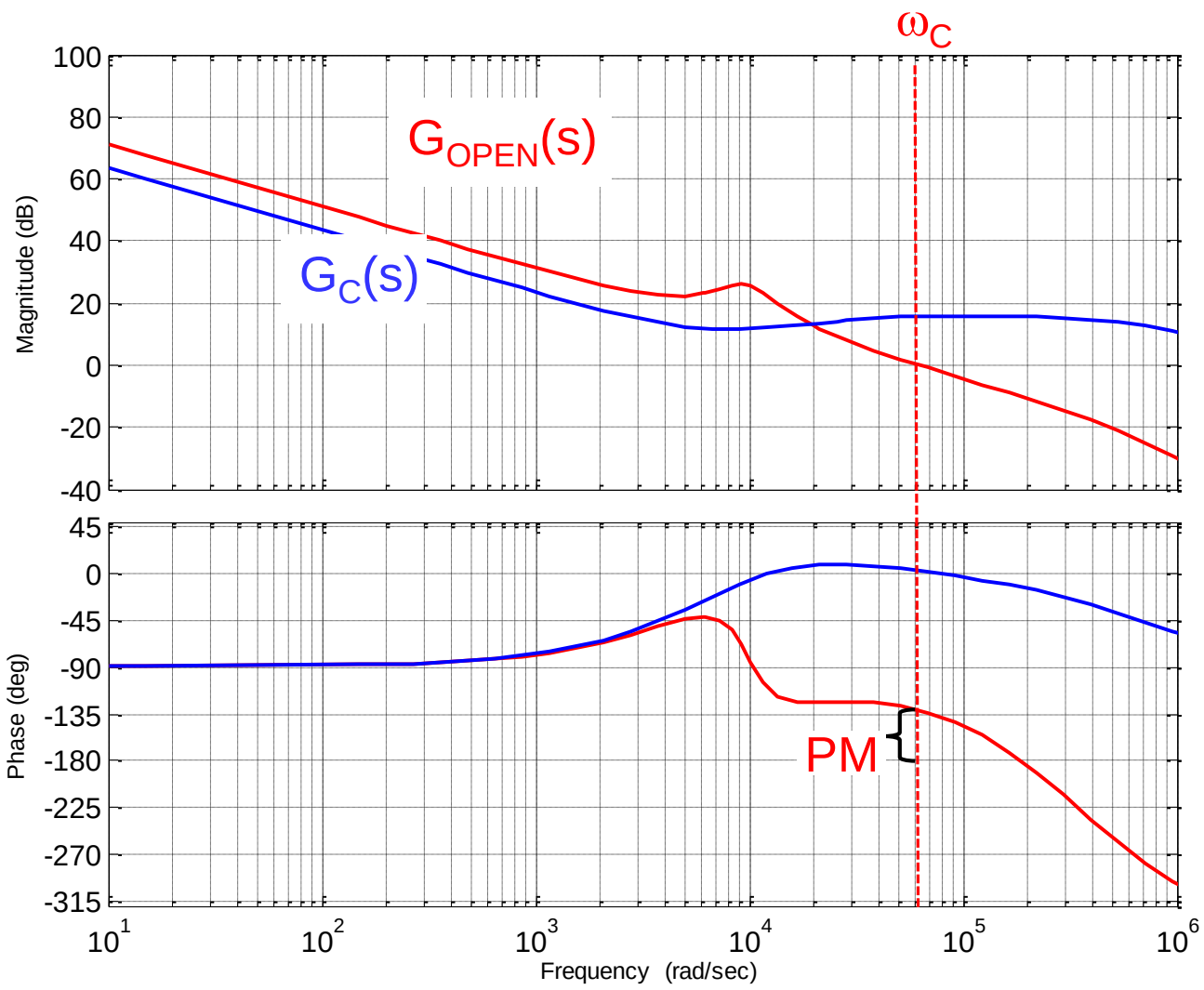
Digital Control Loop Design - Buck Converter Example

- Complete open loop $G_{OPEN}(s)$

$$G_{PWM}(s) \cdot G_{TD}(s) \cdot G_{SYS}(s) \cdot G_{SEN}(s) \cdot G_{ADC}(s) \cdot G_C(s)$$

$$G_C = \frac{4031745(s+7496)(s+7496)}{s(s+2.427 \cdot 10^4)(s+6.283 \cdot 10^5)}$$

- $\omega_C = 62.831 \cdot 10^3$ rad/s
- **PM = 48.7°**



Digital Control Loop Design - Buck Converter Example

- After discretization of $G_c(s), T_s = 5\mu s$, using “Tustin” approximation we get

$$G_C(z) = \frac{3.836 - 3.554z^{-1} - 3.831z^{-2} + 3.559z^{-3}}{1 - 1.664z^{-1} - 0.4669z^{-2} + 0.1966z^{-3}}$$

or

$$G_C(z) = \frac{3.836(z+1)(z-0.9632)(z-0.9632)}{(z-1)(z-0.8856)(z+0.222)}$$

Digital Control Loop Design - Buck Converter Example

- After discretization of $G_c(s), T_s = 5\mu s$, using “Tustin” approximation we get

$$G_C(z) = \frac{3.836 - 3.554z^{-1} - 3.831z^{-2} + 3.559z^{-3}}{1 - 1.664z^{-1} - 0.4669z^{-2} + 0.1966z^{-3}}$$

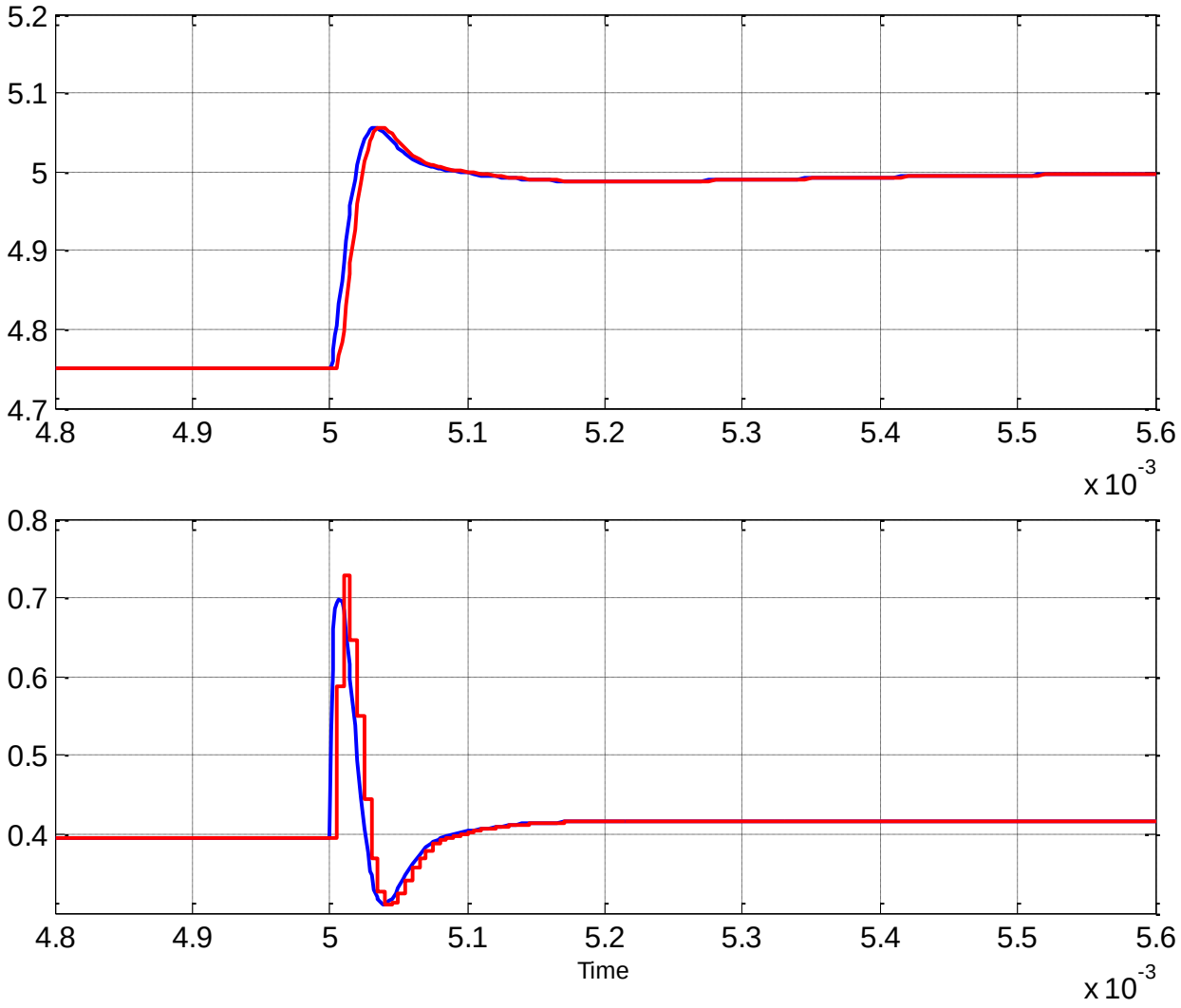
or

$$G_C(z) = \frac{3.836(z+1)(z-0.9632)(z-0.9632)}{(z-1)(z-0.8856)(z+0.222)}$$

Please note
that analog controller
Type III is transformed into
3P3Z digital equivalent

Digital Control Loop Design - Buck Converter Example

- Comparing $G_c(s)$ and $G_c(z)$ responses we can confirm that we have expected performance of digital controller



Digital Control Loop Design – Summary

- Understanding differences between analog and digital control loop are crucial to design stable digital controller
- Incorporating delays caused by digital implementation of control loop into analog control loop allows to utilize all known methods for analog design also for designing digital switched mode power supplies

SOFTWARE & HARDWARE ENABLEMENT FOR POWER CONVERSION APPLICATIONS

NXP Real Time Control Embedded Systems Libraries

- Libraries of s/w algorithms for Math, Motor Control, Power Conversion, Filters and Advanced functions
- Supports Kinetis MCUs, DSCs and key compilers (CW, KDS, IAR, Keil)
- Supports 16-bit, 32-bit, 64-bit fixed point and single precision floating point arithmetic
- Algorithms tested extensively through MATLAB Simulink reference models
- **Free of charge:** Binary code released through NXP website

NXP Real Time Control Embedded Systems Libraries

- **Math Library (MLIB)**
  - Simple math functions (addition, subtraction, multiplication, division, shifts...) with and without saturation. This aim of this block is to have efficient and fast mathematics built on the MCU's instructions.
- **General Function Library (GFLIB)**
  - Contains the basic building blocks of a real-time control application. Functions for basic mathematical calculations, trigonometric functions, simple look-up table and control functions such as PI and PID controllers.
- **General Motor Control Library (GMCLIB)**
  - The fundamental blocks of a motor control application. The libraries include vector modulation, Park and Clarke transformations and specific motor related functions to build digitally controlled motor drives.
- **General Digital Filter Library (GDFLIB)**
  - Includes filter functions for signal conditioning.
- **Advanced Motor Control Library (AMCLIB)**
  - Functions that enable the construction of a variable speed, AC motor drive system that implements field oriented control techniques without position or speed sensors to provide the lowest cost solution.
- **Power Control Library (PCLIB)**
  - Contains control loop algorithms required in Power Control applications.

NXP Real Time Control Embedded Systems Libraries

MLIB

Absolute Value, Negative Value
- 12 functions

Add/Subtract Functions
- 32 functions

Multiply (+ negation, round)
- 52 functions

Multiply + Accumulate (+ round)
- 36 functions

Multiply + Subtract (+ round)
- 36 functions

Division
- 54 functions

Reciprocal
- 8 functions

Shifting
- 32 functions

Count leading bits, Round, Saturation, Sign
- 11 functions

Conversions
- 10 functions

GFLIB

Trigonometric Functions
- Sin, Cos, Tan,Asin, Acos, Atan, AtanYX, atanYXShifted

Limitation Functions
- Limit, VectorLimit LowerLimit, UpperLimit

PI(D) Controller Functions
- Parallel form PI (+ windup + limit)
- Parallel form PID (+ windup + limit)

Interpolation
- Look-up table (periodical, non-periodical)

Hysteresis Function

Signal Integration Function
- Integrator

Signal Ramp Function
- Ramp
- Dynamic ramp
- FlexRamp
- Dynamic flex ramp

Square Root function
- Square root

GDFLIB

Moving Average Filter

Infinite Impulse Filter
- 1st order
- 2nd order
- 3rd order
- 4th order

GMCLIB

Clark Transformation
- Clark, Inverse Clark

Park Transformation
- Park, Inverse Park, ParkInv

Duty Cycle Calculation
- Space vector modulations

Elimination of DC Ripples

Decoupling of PMSM Motors

AMCLIB

Tracking Observers
- Angle Tracking Observer, Tracking Observer
- PMSM BEMF observer in D/Q

PCLIB

Controllers
- PI controller
- PI with LP filter
- PID controller
- 2P2Z controller
- 3P3Z controller

TWR-SMPS-LVFB: Low Voltage SMPS Tower Peripheral Module

- Safe, low voltage hardware for full-bridge converter with synchronous rectification
- Low voltage, low cost modular board for topology used in high voltage, high power applications such as server and telecom power supplies
- On board dynamic load circuit
- Compatible with Kinetis V series MCUs and Digital Signal Controllers (DSCs) Tower MCU Modules
- Input voltage 20-30V DC or AC-DC adapter with 24V @ 3A
- Output power up to 40 watt, output voltage 5V @ 8A
- Cost effective design, safe, robust, easy to use
- Suitable for:
  - Average current mode control implementation
  - Peak current mode control implementation
  - Voltage mode control implementation
- TWR card also can be used to implement Phase shift full bridge topology
- CE/FCC certified

$190 SRP

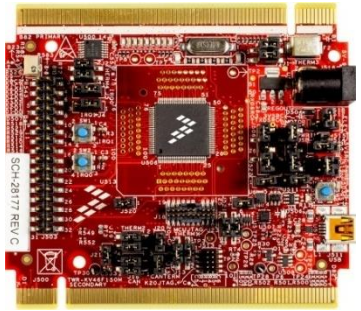
TWR-SMPS-LVFB: Set Up

TWR-SMPS-LVFB

+

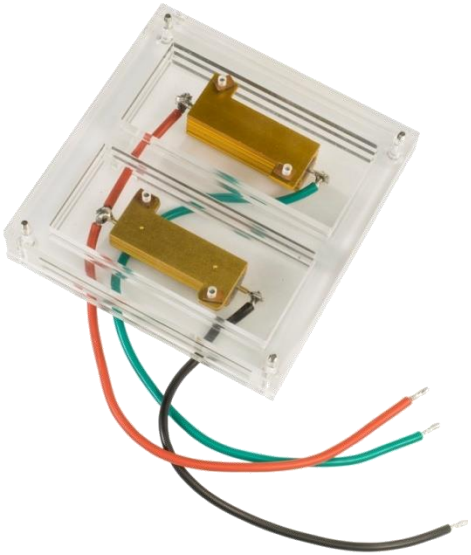
TWR-ELEV

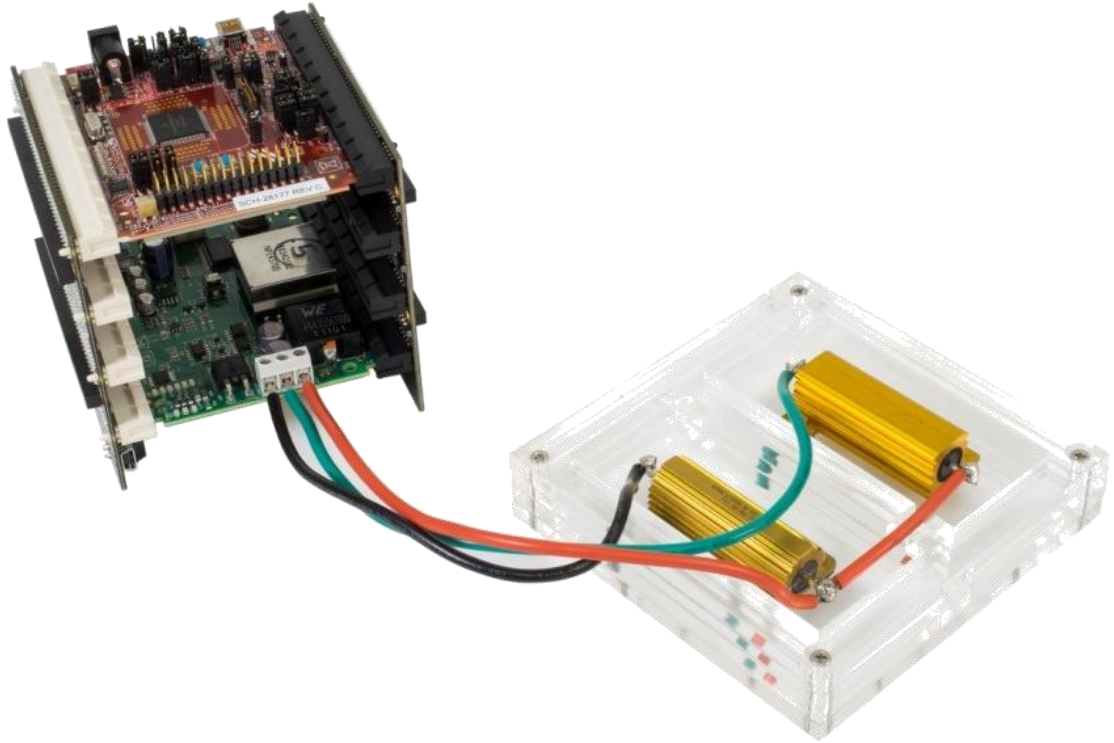
+



TWR-KV46F150M

+





HVP-MC3PH: High Voltage Platform - Overview

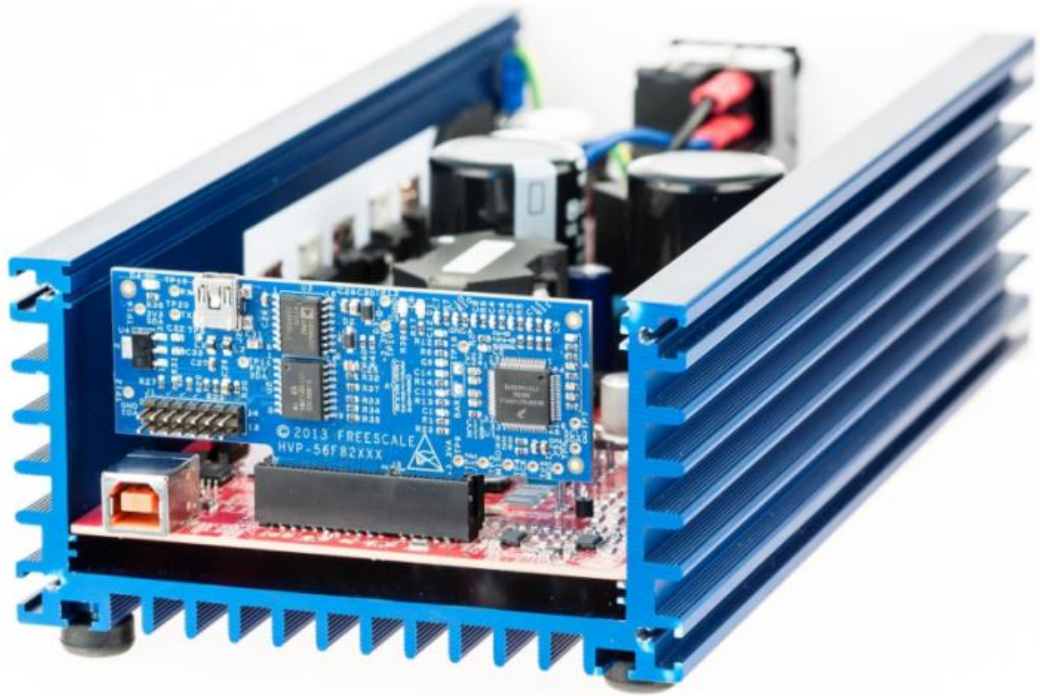
- Hardware and software development platform for 3-phase motor control & Power Factor Correction
- Supports Kinetis V series MCUs and Digital Signal Controllers (DSCs)
- Motor output power up to 1 / 2Hp
- Input voltage 85-240V AC or 110-390VDC
- Safe, robust and easy to use
- Suitable for:
  - PMSM, BLDC and ACIM motors
  - White goods
  - Industrial drives
  - Pumps, fans, compressors
  - Air conditioners
  - Power Factor Correction
- 5KV optical isolation for USB communication and debugging
- CE/FCC certified

www.nxp.com/hvp

ASP: $600 including a KV46 controller card

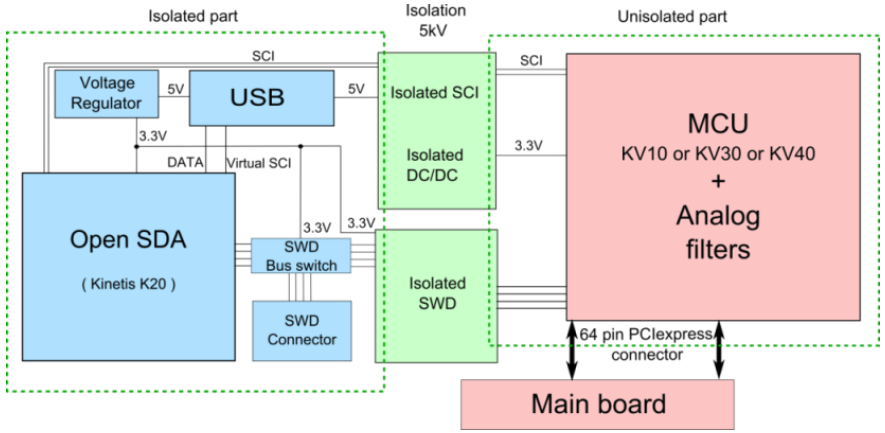
HVP-MC3PH: High Voltage Platform - Features

- Base board with plug-in controller cards for KV1x/KV3x/KV4x MCUs and MF56F82xx DSC
- Electrical Specifications
  - Input voltage 85-240VAC, 110-390VDC
  - Output Current 8 A peak.
  - Output power
    - 1kW without PFC at 250VAC
    - 800W with PFC at >90VAC
- Analog sensing (input voltage, dcb voltage, dcb current, phase currents, back-EMF voltage, PFC currents, IGBT module temperature monitoring)
- Motor speed/position sensors interface (Encoder, Hall, Tacho generator)
- H/W Over-Current Fault Protection
- On-board Interleaved Power Factor Correction
- Over voltage comparator with DC-brake resistor interface
- SCI to USB optically isolated interface
- Current Inrush circuit



HVP-MC3PH Controller Cards: Kinetis KV1x/KV3x/KV4x, DSC

- **HVP-KV10Z32**
  - MKV10Z32VLF7 MCU (ARM CM0+, 75MHz, 32KB)
- **HVP-KV31F120M**
  - MKV31F512VLL12 MCU (ARM CM4, 120MHz, 512KB)
- **HVP-KV46F150M**
  - MKV46F256VLL15 MCU (ARM CM4, 150MHz, 256KB)
- **HVP-56F82748**
  - MC56F82748 DSC (568000EX, 100MHz, 64KB)
- SWD / JTAG isolation up to 5kV
- Open SDA (for Kinetis MCUs only) Interface
- Design optimized for low noise
- On board isolated power supply for safe debugging
- Controller cards can also be used stand-alone
- Cost effective design



HVP-LLC: LLC Resonant Development Kit

- Technical parameters:
  - Input voltage: 390VDC
  - Output power: 250W
  - Output power: 12V
  - Output Current: 21A
  - Switching frequency 75-300kHz
- Main Board Features:
  - High efficiency
  - Replaceable Controller card
  - GaN power FETs
  - Synchronous rectifier
  - Analog sensing (resonant tank current, output voltage, output current, phase currents, back-EMF voltage, PFC currents, IGBT module temperature monitoring)
  - Input/Output over current protection
  - Auxiliary flyback power supply
  - Board designed for easy development and debugging

ASP: ~$499 including a KV46 controller card
Available Q3/2016

SECURE CONNECTIONS
FOR A SMARTER WORLD

ATTRIBUTION STATEMENT