Math Review Handout

Dan Boschen

Last Updated: Sept 14, 2024

Basics

Know and understand the following relationships:

$e^{j\theta} = \cos(\theta) + j\sin(\theta)$

$e^{j\theta}$ is the same as $1\angle\theta$

$360^\circ = 2\pi$ radians

$y = \log(x) \rightarrow 10^y = x$

$y = \ln(x) \rightarrow e^y = x$

$\log(xy) = \log(x) + \log(y)$

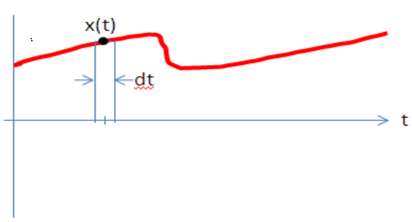
$\log(x^y) = y\log(x)$

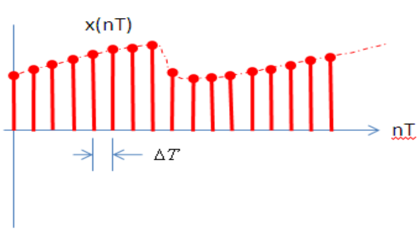
$\log_m(N) = \dfrac{\log_{10}(N)}{\log_{10}(m)}$

Be able to visualize how integration and summation are related:

Both are “area under the curve”

$$\int x(t)dt \cong \sum x(nT)\Delta T$$





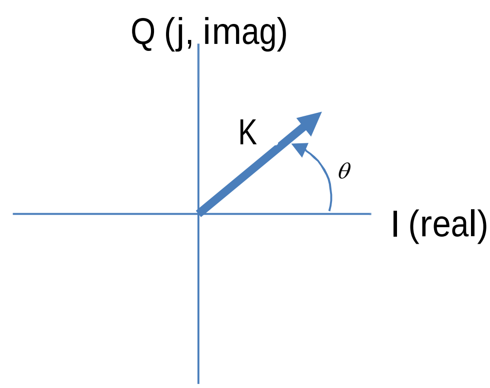
Never Forget:

dB of a **magnitude** is 20log10(magnitude ratio)

dB of a **power** is 10log10(power ratio)

Visualize a vector on a complex plane (phasor):

$V = Ke^{j\theta}$, where K, θ are constants:

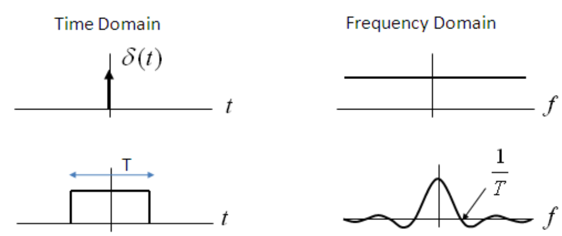


When you multiply vectors- add the phase.

$V_1 = K_1\angle\theta_1, V_2 = K_2\angle\theta_2$

$V_1V_2 = K_1K_2\angle(\theta_1 + \theta_2)$

Memorize the Fourier Transform for a pulse and impulse function



Math Review Handout

Dan Boschen

Last Updated: Sept 14, 2024

Useful Relationships

Series:

Geometric (general):

$$\sum_{n=m}^{N-1} a^n = \frac{a^m - a^N}{1-a}$$

Geometric (finite):

$$\sum_{n=0}^{N-1} a^n = \frac{1-a^N}{1-a}$$

Geometric (infinite):

$$\sum_{n=0}^{\infty} a^n = \frac{1}{1-a}, \quad |a| < 1$$

$$\sum_{n=0}^{\infty} a^{-n} = \frac{a}{a-1}, \quad |a| > 1$$

Ramp:

$$\sum_{n=0}^{\infty} n a^n = \frac{a}{(1-a)^2}, \quad |a| < 1$$

Taylor: $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$

Approximations

From Taylor Series:

$e^x \approx 1 + x \quad$ for x << 1

From Taylor Series:

$$e^x = \frac{e^{x/2}}{e^{-x/2}} \approx \frac{1+x/2}{1-x/2} \quad \text{for x} << 1$$

Binomial:

$(1+x)^n \approx 1 + nx \quad$ for x << 1

Binomial: $\frac{1}{1+x} \approx 1 - x \quad$ for x << 1

Small Angle:

$\sin(x) \approx x \quad$ for x << 1 (rad)

Identities:

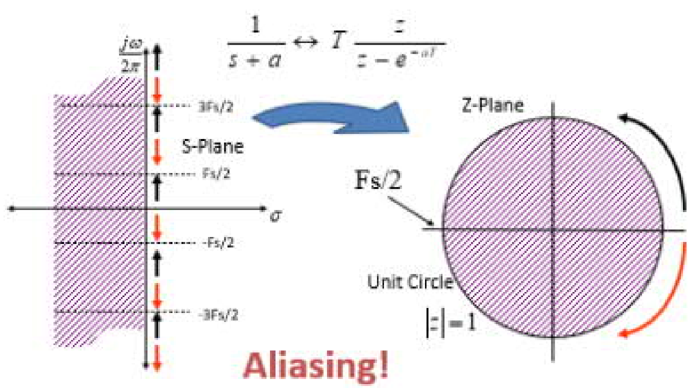
$$\cos(\phi)\cos(\theta) = \frac{\cos(\phi-\theta) + \cos(\phi+\theta)}{2}$$

$$\sin(\phi)\sin(\theta) = \frac{\cos(\phi-\theta) - \cos(\phi+\theta)}{2}$$

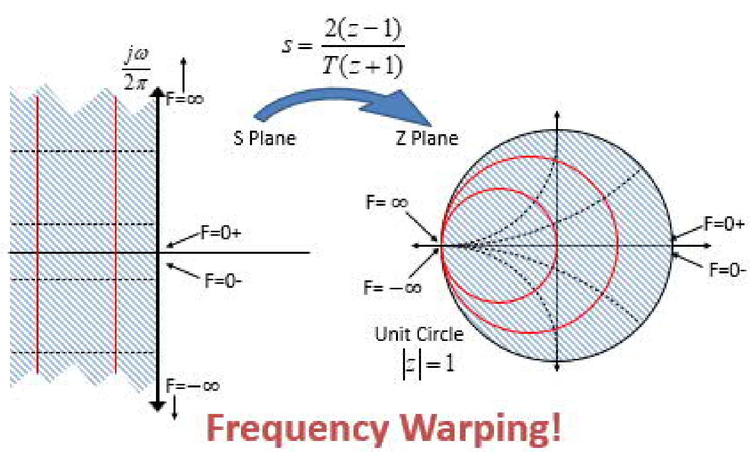
$$\sin(\phi)\cos(\theta) = \frac{\sin(\phi+\theta) + \sin(\phi-\theta)}{2}$$

Converting s to z

Method of Impulse Invariance (if all poles)



Method of Bilinear Transform



Math Review Handout

Dan Boschen

Last Updated: Sept 14, 2024

Transforms (All time and sample domain functions causal and stable)

Time Domain	Laplace	Sample Domain	Z
$f(t)$	$F(s)=\int_0^\infty f(t)e^{-st}dt$	$f[nT]$	$F(z)=\sum_{n=0}^{\infty} f[nT]z^{-n}$
Differentiation $\frac{df}{dt}$	$sF(s)-f(0)$	Difference $f[n]-f[n-1]$	$\frac{z-1}{z}F(z)$
Integration $\int_0^t f(\tau)d\tau$	$\frac{1}{s}F(s)$	Accumulation $\sum_{n=0}^{n} f[n]$	$\frac{z}{z-1}F(z)$
Exponential Decay $e^{-at}f(t)$	$F(s+a)$	Geometric Decay $a^{-n}f[n]$	$F(za)$
Time Delay $f(t-a)$	$e^{-sa}F(s)$	Sample Delay $f[n-m]$	$z^{-m}F(z)$
Impulse $\delta(t)$	1	Unit Sample $\delta[n]$	1
1	$\frac{1}{s}$	1	$\frac{z}{z-1}$
t	$\frac{1}{s^2}$	nT	$\frac{Tz}{(z-1)^2}$
e^{at}	$\frac{1}{s-a}$	e^{anT}	$\frac{z}{z-e^{aT}}$
		a^{nT}	$\frac{z}{z-a}$
Initial Value Theorem	$\lim_{t\to 0^+} = \lim_{s\to\infty} sF(s)$	Initial Value Theorem	$f[0]=\lim_{z\to\infty} F(z)$
Final Value Theorem	$\lim_{t\to\infty} = \lim_{s\to 0} sF(s)$ *(all poles in LHP, no more than one pole at the origin)*	Final Value Theorem	$f[n]=\lim_{n\to\infty} = \lim_{z\to 1}(z-1)F(z)$ *(all poles in unit circle, no more than one pole at z=1)*
$f(t)*g(t)$	$F(s)G(s)$	$f[n]*g[n]$	$F(z)G(z)$

"All assumed causal and stable":
Time Domain
f(t)=0 for t<0, all poles in LHP, ROC contains jw axis and positive infinity
Sample Domain
f(n)=0 for n<0, all poles inside unit circle, ROC contains unit circle and infinity

Math Review Handout

Dan Boschen

Last Updated: Sept 14, 2024

PYTHON USEFUL COMMANDS

np: import numpy as np
sig: import scipy.signal as sig
con: import control as con

Polynomial factoring and manipulation

np.roots()	Polynomial roots, ex: np.roots([1, 6, 10]) to solve for roots of x^2+6x+100 (-3 ± j)
np.poly()	Polynomial from roots, ex: np.poly([-3+1j, -3-1j]) = [1., 6., 10.]
np.convolve()	Convolve (multiply) two polynomials, ex np.convolve([1, 5],[1, 2, 4]) for (x+5)(x^2+2x+4)
np.polydiv()	Deconvolve (divide) two polynomials
sig.residue()	Partial-fraction expansion

Converting between s and z

sig.bilinear()	Bilinear transform from s to z, in either zero-pole-gain or transfer function (TF) form
con.c2d()	Continuous to discrete mapping s to z. methods = 'zoh' (zero order hold), 'foh' (first order hold), 'impulse' (impulse invariance), 'tustin' (Bilinear transform)

Control systems (see [https://python-control.readthedocs.io/en/0.9.1/)](https://python-control.readthedocs.io/en/0.9.1/)

[num,den]= con.zpk2tf(z,p,k)	zero-pole to transfer function conversion
[z,p,k]= con.tf2zpk([num],[den])	Transfer function to zero-pole conversion

con.sys=tf([*num*],[*den*],*TS*) Transfer function system data structure, TS = sampling interval (omitted for a continuous system.)

Example:	sys=tf(3,[1 2])	for system 3/(s+2)
Example:	sys=tf(3,[1 2],1)	for system 3/(z+2), sampling time normalized or 1 second

All of the following commands operate on sys entered as above.

con.nyquist(sys)	Nyquist Plot
con.rlocus(sys)	Root Locus Plot
con.pzmap(sys)	Map of poles and zeros
con.bode(sys)	Bode Plot
con.step_response(sys)	Step Response
con.impulse_response(sys)	Impulse Response
con.feedback(sys1,sys2)	Closed loop response from open loop response
con.minreal(sys)	Reduce transfer function (good practice to always use this!)

Math Review Handout

Dan Boschen
Last Updated: Sept 14, 2024

MATLAB/OCTAVE USEFUL COMMANDS

Polynomial factoring and manipulation

roots()	Polynomial roots, ex: roots([1 6 10]) to solve for roots of x^2+6x+100 (-3 ± j)
poly()	Polynomial from roots, ex: poly([-3+j -3-j]) = [1 6 10]
conv()	Convolve (multiply) two polynomials, ex conv([1 5],[1 2 4]) for (x+5)(x^2+2x+4)
deconv()	Deconvolve (divide) two polynomials
residue()	Partial-fraction expansion

Converting between s and z

bilinear()	Bilinear transform from s to z, in either zero-pole-gain or transfer function (TF) form
impinvar()	(MATLAB ONLY) Impulse invariance from s to z, in either zero-pole-gain or TF form
c2d()	Continuous to discrete, supports Bilinear and Matched-Z transform (Matlab also supports impulse invariance within the c2d command)

Control systems

[num,den]= zp2tf(z,p,k)	zero-pole to transfer function conversion
[z,p,k]= tf2zp([num],[den])	Transfer function to zero-pole conversion

sys=tf([*num*],[*den*],*TS*) — Transfer function system data structure, TS = sampling interval (omitted for a continuous system.)

Example: sys=tf(3,[1 2])	for system 3/(s+2)
Example: sys=tf(3,[1 2],1)	for system 3/(z+2), sampling time normalized or 1 second

All of the following commands operate on sys entered as above.

nyquist(sys)	Nyquist Plot
rlocus(sys)	Root Locus Plot
pzmap(sys)	Map of poles and zeros
bode(sys)	Bode Plot
step(sys)	Step Response
impulse(sys)	Impulse Response
feedback(sys1,sys2)	Closed loop response from open loop response
minreal(sys)	Reduce transfer function (good practice to always use this!)