

Tuesdays &
Thursdays 12:00-
13:30 ET
GGBL 2147

Lecture 3

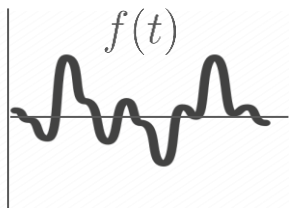


ME599-004: Data-Driven Methods for Control Systems

Winter 2024

Instructor: Uduak (*Who-dwak*) Inyang-Udoh

Fourier Series & Transforms



$$f(t) = \sum_{k=-\infty}^{\infty} c_k e^{ik\pi t/L}$$

Fourier series

$$\langle f, g \rangle = \int f \bar{g}$$

$$\langle g, f \rangle = \int g \bar{f}$$

$$\langle f, \bar{g} \rangle = \langle g, f \rangle$$

$$c = \frac{1}{2L} \langle f(t), e^{i\kappa\pi t/L} \rangle$$

$$= \frac{1}{2L} \int_{-L}^L f(t) e^{-i\kappa\pi t/L} dt$$

$$\omega = \kappa\pi/L, \Delta\omega = \pi/L$$

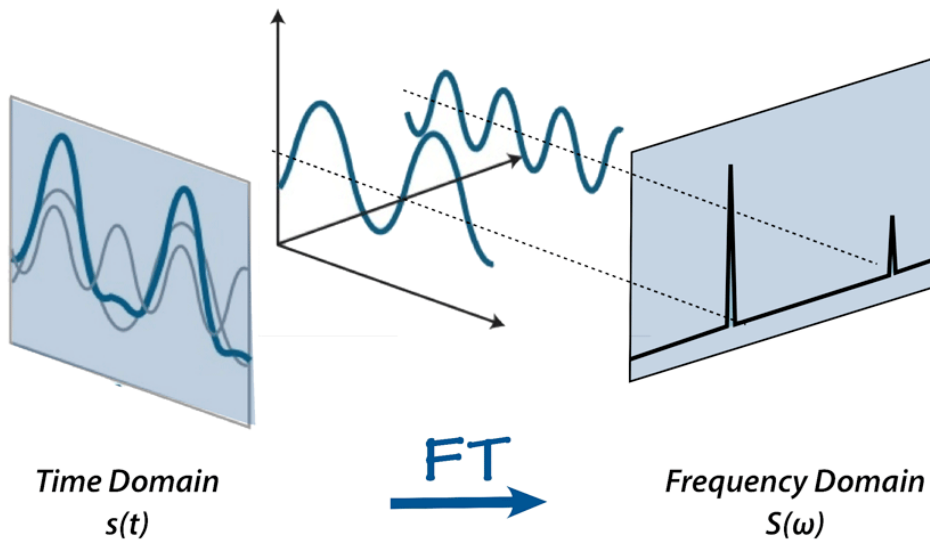
$$f(t) = \lim_{\Delta\omega \rightarrow 0} \sum_{k=-\infty}^{\infty} \frac{\Delta\omega}{2\pi} \langle f(t), e^{i\omega t} \rangle_{[-\frac{\pi}{\Delta\omega}, \frac{\pi}{\Delta\omega})} e^{ik\Delta\omega t}$$

\ c_k

$$\hat{f}(\omega) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

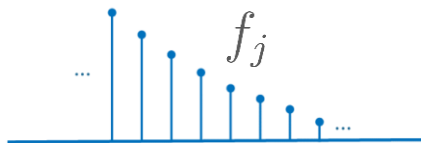
$$f(t) = \mathcal{F}^{-1}(\hat{f}(\omega)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t} d\omega$$

Fourier Series & Transforms



[Source](#)

Discrete & Fourier Transforms



$$\omega_N = e^{-i\frac{2\pi}{N}}$$

$$\hat{f}_k = \sum_{j=0}^{N-1} f_j e^{-ik\frac{2\pi}{N}j}$$

$\frac{1}{\sqrt{N}}$

$$f_j = \frac{1}{N} \sum_{k=0}^{N-1} \hat{f}_k e^{ij\frac{2\pi}{N}k}$$

$\frac{1}{\sqrt{N}}$

$$\begin{array}{c}
 j \\
 0 \quad 1 \quad 2 \quad \dots \quad N-1 \\
 k \\
 \begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{N-1} \end{bmatrix}
 \end{array}
 \stackrel{\frac{1}{\sqrt{N}}}{=}
 \begin{bmatrix}
 1 & 1 & 1 & \dots & 1 \\
 1 & \omega_N^1 & \omega_N^2 & \dots & \omega_N^{N-1} \\
 1 & \omega_N^2 & \omega_N^4 & \dots & \omega_N^{2(N-1)} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 1 & \omega_N^{N-1} & \omega_N^{2(N-1)} & \dots & \omega_N^{(N-1)^2}
 \end{bmatrix}
 \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{N-1} \end{bmatrix}$$

DFT Matrix

Fast Fourier Transforms



$$\hat{f}_k = \sum_{j=0}^{N-1} f_j e^{-ik \frac{2\pi}{N} j} \quad \leftarrow \text{DFT} \quad \hat{f}_{k+N} = \hat{f}_k$$

$$\hat{f}_k = \sum_{j=0}^{N/2-1} \underbrace{f_{2j}}_{\text{even}} e^{-ik \cdot \frac{2\pi}{N/2} j} + e^{-ik 2\pi/N} \sum_{j=0}^{N/2-1} \underbrace{f_{2j+1}}_{\text{odd}} e^{-ik \frac{2\pi}{N} 2j}$$

$$\hat{f}_{k+\frac{N}{2}} = \sum_{j=0}^{N/2-1} \underbrace{f_{2j}}_{\text{even}} e^{-ik \cdot \frac{2\pi}{N/2} j} - e^{-ik 2\pi/N} \sum_{j=0}^{N/2-1} \underbrace{f_{2j+1}}_{\text{odd}} e^{-ik \frac{2\pi}{N} 2j}$$

Algorithm 1: fft

Data: Signal vector $\mathbf{f} = \{f_0, \dots, f_{N-1}\}$

Result: $\hat{\mathbf{f}} = \{\hat{f}_0, \dots, \hat{f}_{N-1}\}$

if $N = 1$ **then**

$\hat{\mathbf{f}} \leftarrow \mathbf{f}$

else

$\hat{\mathbf{f}}_{\text{even}} \leftarrow \text{fft}(\{f_0, f_2, \dots, f_{N-1}\});$

$\hat{\mathbf{f}}_{\text{odd}} \leftarrow \text{fft}(\{f_1, f_3, \dots, f_{N-2}\});$

$\mathbf{n} \leftarrow \{1, 2, \dots, \frac{N}{2} - 1\};$

$\omega \leftarrow e^{i \frac{2\pi}{N} \mathbf{n}};$

$\hat{\mathbf{f}} \leftarrow [\hat{\mathbf{f}}_{\text{even}} + \omega \hat{\mathbf{f}}_{\text{odd}}, \hat{\mathbf{f}}_{\text{even}} - \omega \hat{\mathbf{f}}_{\text{odd}}]$

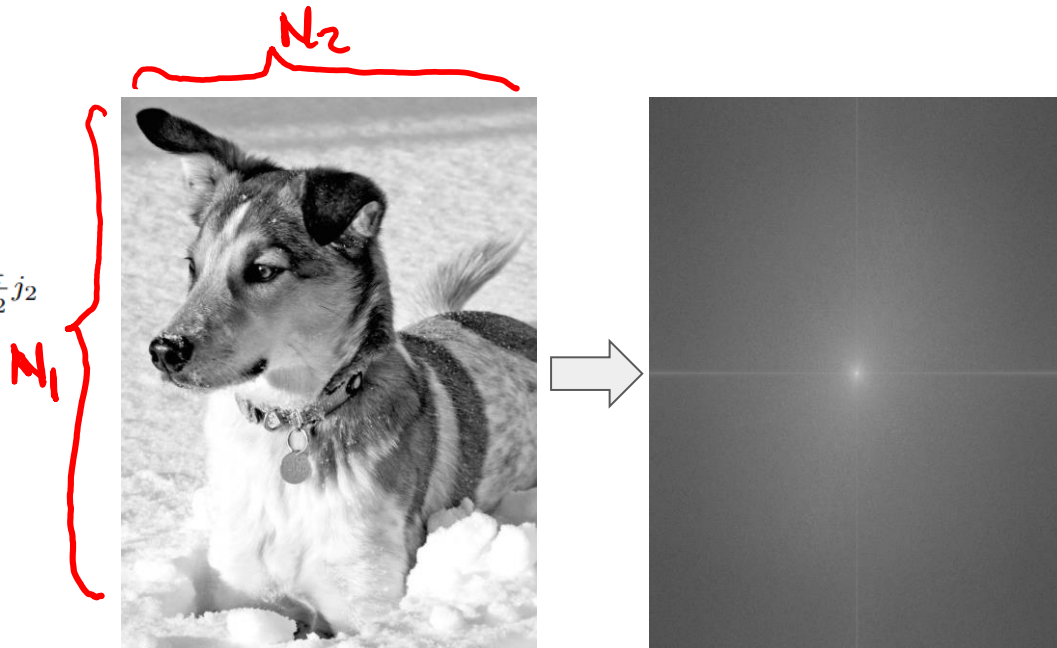
end

2D Fourier Transform

Fourier transforms do not only apply to temporal signals; applies to spatial signals

$$\hat{f}_k = \sum_{j=0}^{N-1} f_j e^{-ik \frac{2\pi}{N} j}$$

$$\hat{f}_{k_1, k_2} = \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} f_{j_1, j_2} e^{-ik_1 \frac{2\pi}{N_1} j_1} e^{-ik_2 \frac{2\pi}{N_2} j_2}$$



1. Filtering
 - a. Audio
 - b. Images
2. Convolution:
 - a. LTI systems
 - b. Image processing
3. Differential Equations: (Laplace Transform also used)
4. Compression: DCT often used
 - a. Audio
 - b. Images
 - c. Media
5. Time Series Analysis: trend analysis, anomaly detection
 - a. audio signals
 - b. finance
 - c. Healthcare e.g. EEG
6. Others: Feature engineering/ Data augmentation, CNN's, Natural Language Processing

Noise Filtering

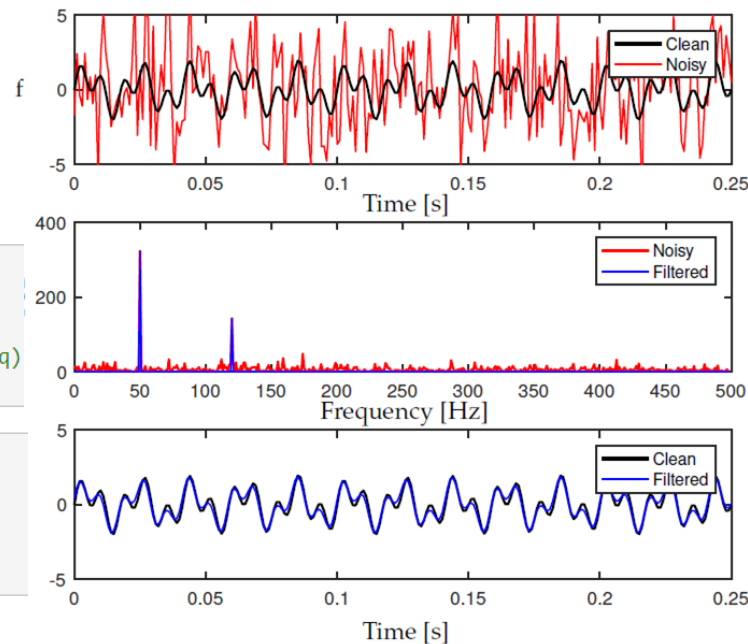
$$f(t) = \sin(2\pi f_1 t) + \sin(2\pi f_2 t)$$

$$f_1 = 50; f_2 = 120$$

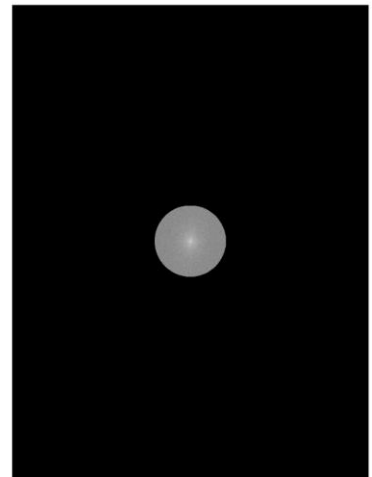
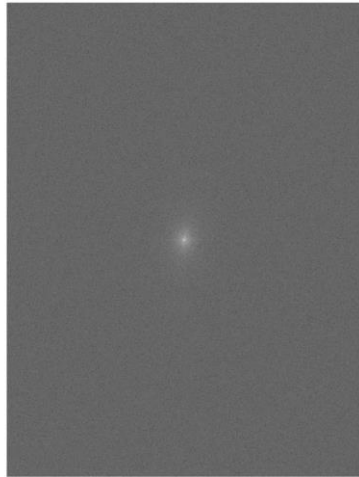
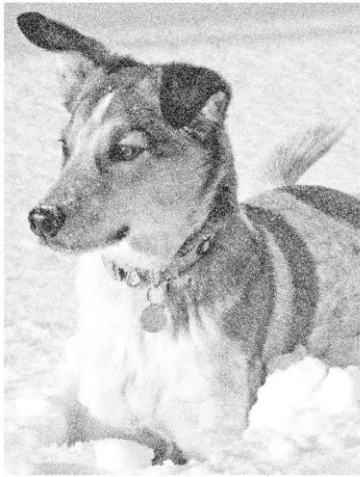
```
% Code is partly courtesy of Brunton and Kutz (2023) text
dt = .001;
t = 0:dt:1;
f = sin(2*pi*50*t) + sin(2*pi*120*t); % Sum of 2 frequencies
f_noisy = f + 2.5*randn(size(t)); % Add some noise
```

```
%% Compute the Fast Fourier Transform FFT
n = length(t);
fhat_noisy = fft(f_noisy,n); % Compute the fast Fourier transform
PSD_noisy = fhat_noisy.*conj(fhat_noisy)/n; % Power spectral density (power per freq)
L = 1:floor(n/2); % Only plot the first half of freqs
```

```
%% Use the PSD to filter out noise
indices = PSD_noisy>100; % Find all freqs with large power
PSDclean = PSD_noisy.*indices; % Zero out all others
fhat_noisy = indices.*fhat_noisy; % Zero out small Fourier coeffs. in fhat
ffilt = ifft(fhat_noisy); % Inverse FFT for filtered time signal
```



Noise Filtering (Cont'd)



Compression

```
figure
Bt=fft2(B);
Btsort = sort(abs(Bt(:))); % Sort by magnitude

% Zero out all small coefficients and inverse transform
percentvec = [.1 .05 .01 .002];
for k=1:4
    keep = percentvec(k);
    thresh = Btsort(floor((1-keep)*length(Btsort)));
    ind = abs(Bt)>thresh;
    Atlow = Bt.*ind; % Threshold small indices
    Flow = log(abs(fftshift(Atlow))+1); % put FFT on log-scale
end
```



.2%



0-5%



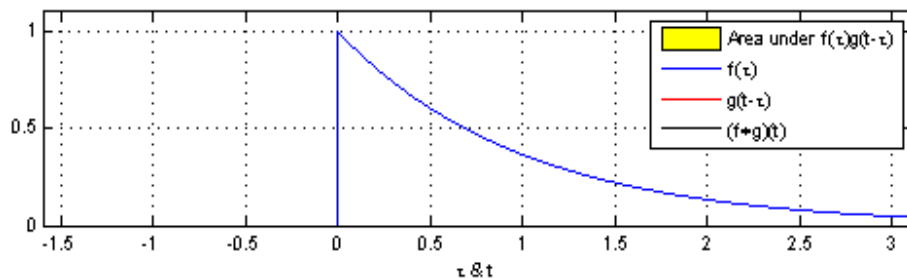
1%

Convolution



$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$$

To convolve a kernel with an input signal:
flip the signal, move to the desired time,
and accumulate every interaction with the kernel



[Source](#)

Fourier Transform of Convolution

$$\mathcal{F}(f * g) = \mathcal{F}(f)\mathcal{F}(g)$$

$$\mathcal{F}^{-1}(\hat{f}\hat{g}) = f * g$$

Proof (1):

$$\begin{aligned}\mathcal{F}(f * g) &= \int_{-\infty}^{\infty} f(\tau) g(t - \tau) \int_{-\infty}^{\infty} e^{-i\omega t} dt d\tau \\ &= \int_{-\infty}^{\infty} f(\tau) e^{-i\omega\tau} d\tau \int_{-\infty}^{\infty} g(t - \tau) e^{-i\omega(t - \tau)} dt \\ &= \int_{-\infty}^{\infty} f(\tau) e^{-i\omega\tau} d\tau \int_{-\infty}^{\infty} g(t'') e^{-i\omega t''} dt'' \\ &= \mathcal{F}(f)\mathcal{F}(g)\end{aligned}$$

Convolution

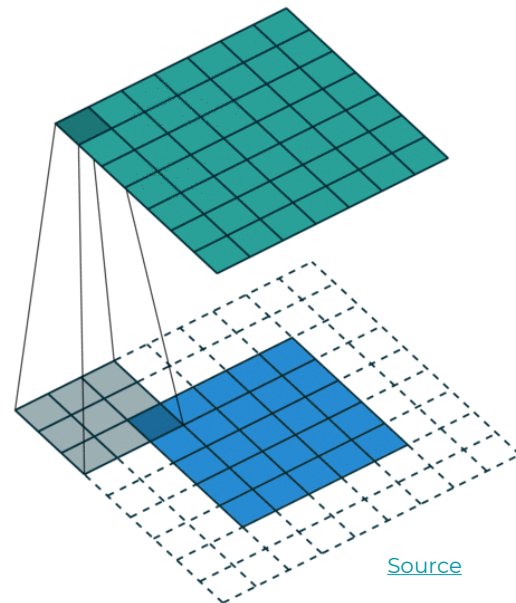
Discrete Convolution

$$(f * g)[n] = \sum_{k=-\infty}^{\infty} f[k]g[n - k]$$

- Edge detection
- Blurring
- Sharpening
- Neural networks

Discrete 2D Convolution

$$\begin{aligned}(F * G)[n_1, n_2] \\ = \sum_{k_2=-\infty}^{\infty} \sum_{k_1=-\infty}^{\infty} F[k_1, k_2] \cdot G[n_1 - k_1, n_2 - k_2]\end{aligned}$$



[Source](#)

Laplace Transform

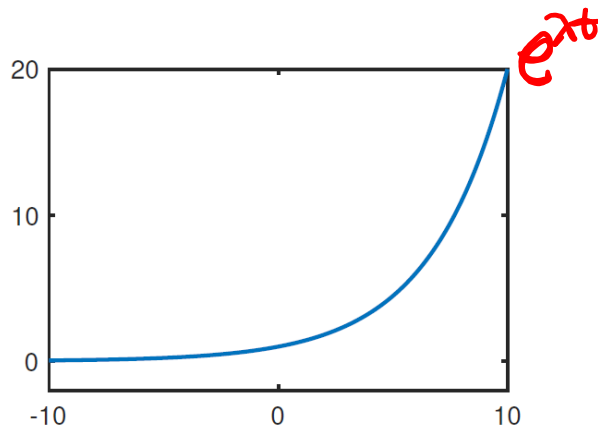
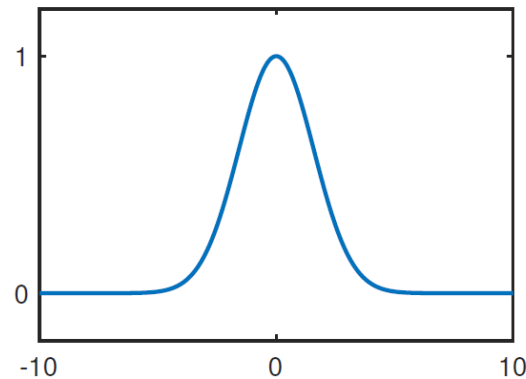


- ODE to algebraic equations
- Control theory

Laplace Transform

- ODE to algebraic equations
- Control theory

$$\hat{f}(\omega) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

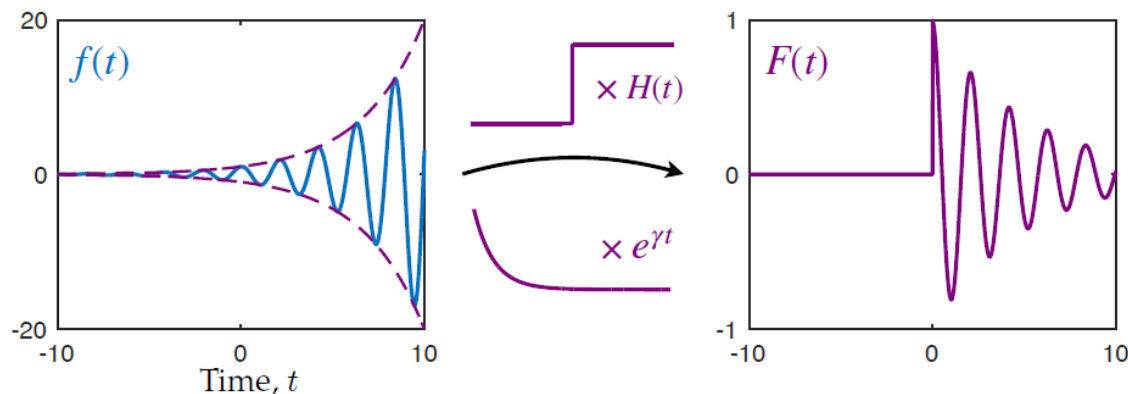


Laplace Transform

- ODE to algebraic equations
- Control theory

$$\hat{f}(\omega) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$

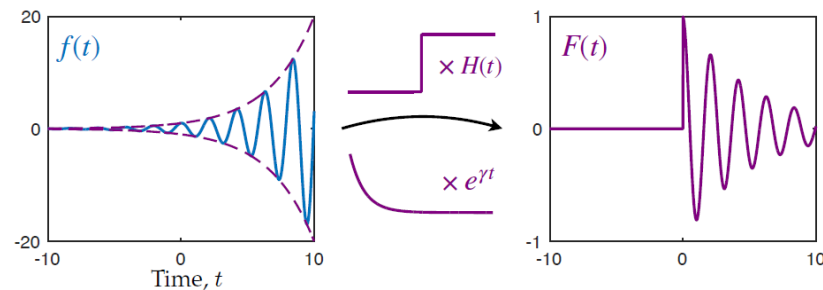
$$F(t) = f(t)e^{-\gamma t}H(t) = \begin{cases} 0 & \text{for } t \leq 0, \\ f(t)e^{-\gamma t} & \text{for } t > 0. \end{cases}$$



Laplace Transform

- ODE to algebraic equations
- Control theory

$$\hat{f}(\omega) = \mathcal{F}(f(t)) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt$$



$$F(t) = f(t)e^{-\gamma t}H(t) = \begin{cases} 0 & \text{for } t \leq 0, \\ f(t)e^{-\gamma t} & \text{for } t > 0. \end{cases}$$

$$\begin{aligned} \hat{F}(\omega) &= \mathcal{F}(F(t)) = \int_{-\infty}^{\infty} F(t)e^{-i\omega t} dt = \int_0^{\infty} f(t)e^{-\gamma t}e^{-i\omega t} dt \\ &= \int_0^{\infty} f(t)e^{-(\gamma+i\omega)t} dt = \int_0^{\infty} f(t)e^{-st} dt = \bar{f}(s) \end{aligned}$$

Laplace Transform



Inverse Laplace Transform

$$F(t) = \mathcal{F}^{-1}(\hat{F}(\omega)) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{F}(\omega) e^{i\omega t} d\omega$$

Multiply both sides by $e^{\gamma t}$

$$\begin{aligned} \textcircled{f(t)H(t)} = e^{\gamma t} F(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{\gamma t} \hat{F}(\omega) e^{i\omega t} d\omega \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{F}(\omega) e^{(\gamma+i\omega)t} d\omega \end{aligned}$$

$$f(t)H(t) = \frac{1}{2\pi i} \int_{\gamma-i\infty}^{\gamma+i\infty} \bar{f}(s) e^{st} ds$$

$$s = \gamma + i\omega$$

$\omega \rightarrow$ in terms of s

$$ds = i d\omega$$

$$\Rightarrow d\omega = \left(\frac{1}{i}\right) ds$$

Laplace Transform



$$\mathcal{L}\left(\frac{d^n}{dt^n} f(t)\right) = -f^{(n-1)}(0) - s f^{(n-2)}(0) - \dots - s^{n-1} f(0) + s^n \bar{f}(s)$$

$$\begin{aligned} \mathcal{L}\left(\frac{d}{dt} f(t)\right) &= \int_0^\infty \underbrace{f'(t)}_{dv} \underbrace{e^{-st}}_u dt & \int dv u &= uv - \int u dv \\ &= \left[e^{-st} f(t) \right]_0^\infty - \int_0^\infty f(t) [-s e^{-st}] dt \\ &= 0 - f(0) + s \int_0^\infty f(t) e^{-st} dt \\ &= -f(0) + s \mathcal{L}(f(t)) \end{aligned}$$

- Brunton and Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*, 2nd ed., Cambridge University Press, 2023