ME 360, Fall 2024 **PROBLEM SET 1** **Due on 3PM Friday, September 6**

Reading: Review your ME 340 notes. $u_0(t)$ denotes the unit step. $\delta(t)$ denotes the Dirac delta.

Problems:

1. Using transform tables, find the Laplace transform of the following:

1. $e^{-2t}u_0(t)$
2. $\cos(2t+\pi)u_0(t)$
3. $te^{-2t}u_0(t)$
4. $(t^3 - u_0(t))u_0(t)$
5. $\sin^3(t)u_0(t)$

2. Using transform tables, find the inverse Laplace transform of the following:

1. $\frac{1}{s}$
2. $\frac{1}{s+2}$
3. $\frac{1}{(s+2)^2}$
4. $\frac{1}{s^2+1}$
5. $\frac{s-1}{(s+3)^2}$

3. Consider the system

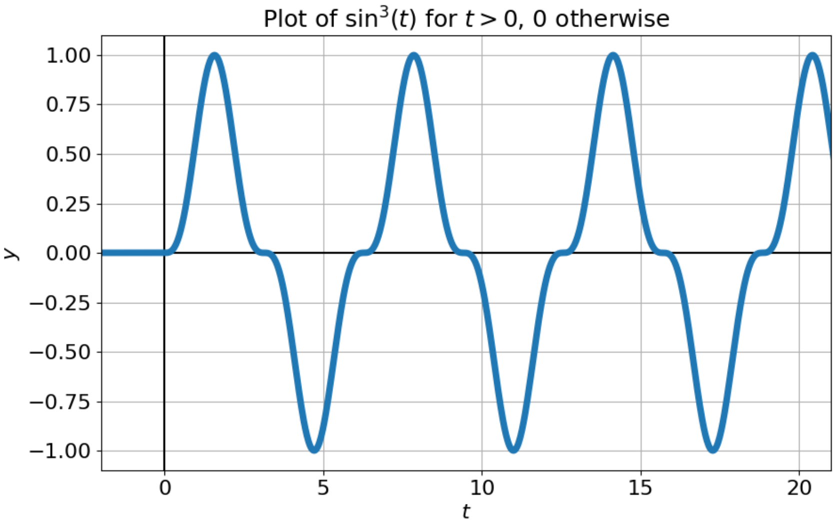
$$\ddot{y}(t) + y(t) = \sin(t)u_0(t) + \delta(t)$$

Use the Laplace transform to solve for $y(t)$ with the given initial condition $y(0-) = 1$ and $\dot{y}(0-) = 0$. What is the significance of $0-$ in setting the initial condition?

4. [Coding Problem] This problem is designed to help you develop and exercise coding skills in the Python programming language. An example Python code is provided and can be accessed on GitHub as well. The code can be run to plot the time signal from Problem 1.5.

As part of this exercise, you are required to suitably modify the code to plot the two time signals from Problems 2.4 and 2.5. These are two time-domain signals that you obtained from calculating the inverse Laplace transform.

The two signals should be plotted on the same figure. Submit your plot with proper axis labels, axis limits, tick marks, and legends indicating which time signals are being plotted. Consult with TA-s during the office hours if you need additional help with Python.



```
import numpy as np
import matplotlib.pyplot as plt

FONTSIZE = 16

def signal(time):
    y = np.zeros(time.shape)
    for i, t in enumerate(time):
        if t >= 0:
            y[i] = (np.sin(t))**3
    return y

t = np.linspace(-3, 22, 1000)
y = signal(t)

plt.figure(figsize=(10, 6))
plt.title(
    'Plot of $\\text{sin}^3(t)$ for $t > 0$, 0 otherwise',
    fontsize=FONTSIZE+2
)
plt.xlabel('$t$', fontsize=FONTSIZE)
plt.ylabel('$y$', fontsize=FONTSIZE)
plt.grid(True)
plt.axhline(y=0, color='black', linestyle='-', alpha=1)
plt.axvline(x=0, color='black', linestyle='-', alpha=1)
plt.plot(t, y, linewidth=5)
plt.xlim(-2, 21)
plt.tick_params(axis='both', which='major', labelsize=FONTSIZE)
plt.show()
```