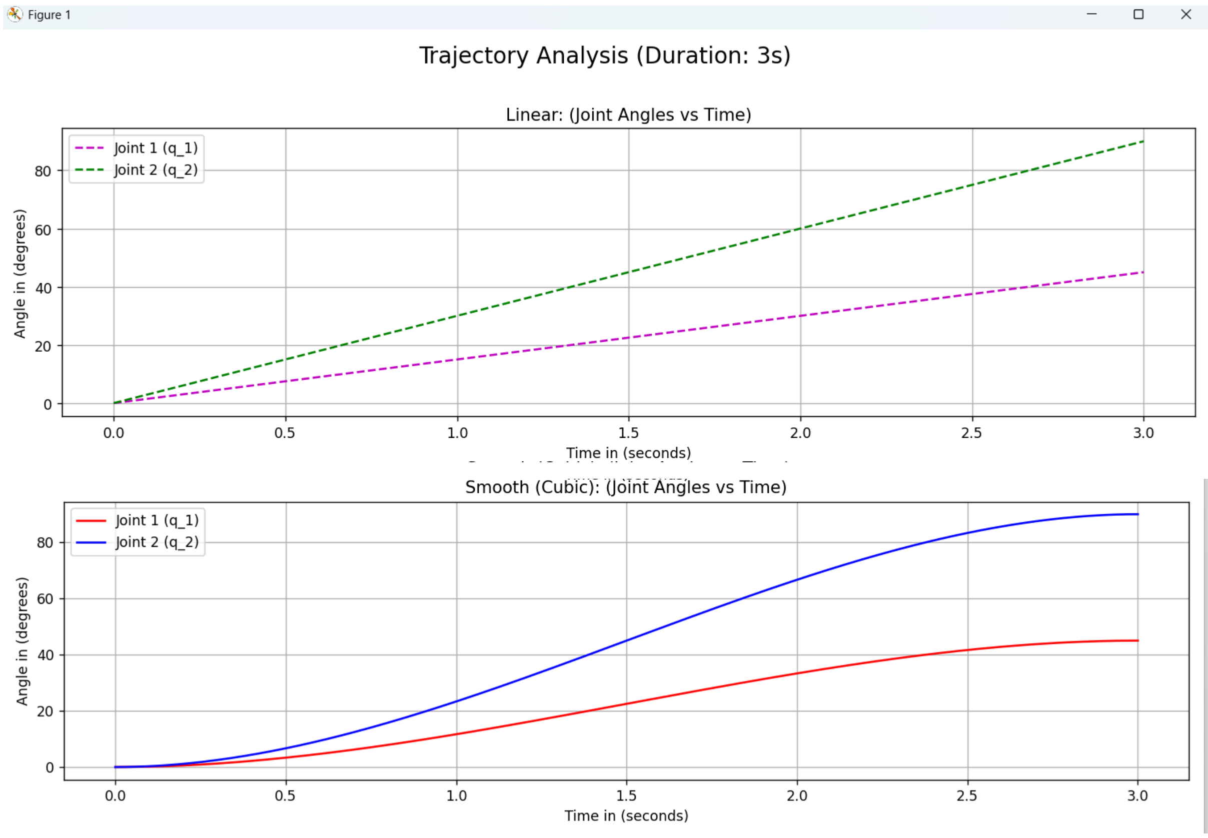
```python
import numpy as np
import matplotlib.pyplot as plt

q_start = np.array([0, 0])
q_end   = np.array([45, 90])
T       = 3
dt      = 0.01

t= np.arange(0, T + dt, dt)

def lin_traj(t_arr, start, end, dur):
    #interpolation
    q_traj = start + (end - start) * (t_arr[:, None] / dur)
    return q_traj

def cub_traj(t_arr, start, end, dur):
    # coefficients
    a0 = start
    a1 = 0
    a2 = 3 * (end - start) / (dur**2)
    a3 = -2 * (end - start) / (dur**3)

    t = t_arr[:, None]
    q_traj = a0 + a1*t + a2*(t**2) + a3*(t**3)

    return q_traj

lin_pos = lin_traj(t, q_start, q_end, T)
cub_pos= cub_traj(t, q_start, q_end, T)

fig, axs = plt.subplots(2, figsize=(12, 10))
fig.suptitle(f'Trajectory Analysis (Duration: {T}s)', fontsize=16)

# plot 1: linear position
axs[0].plot(t, lin_pos[:,0], 'm--', label='Joint 1 (q_1)')
axs[0].plot(t, lin_pos[:,1], 'g--', label='Joint 2 (q_2)')
axs[0].set_title("Linear: (Joint Angles vs Time)")
axs[0].set_ylabel("Angle in (degrees)")
axs[0].set_xlabel("Time in (seconds)")
axs[0].grid(True)
axs[0].legend()

# plot 2: smooth position
```

```
axs[1].plot(t, cub_pos[:,0], 'r-', label='Joint 1 (q_1)')
axs[1].plot(t, cub_pos[:,1], 'b-', label='Joint 2 (q_2)')
axs[1].set_title("Smooth (Cubic): (Joint Angles vs Time)")
axs[1].set_xlabel("Time in (seconds)")
axs[1].set_ylabel("Angle in (degrees)")
axs[1].grid(True)
axs[1].legend()

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```



1.- Differences between linear and smooth trajectories -

Linear trajectory shows the joint angles changing at a constant rate. While mathematically this is the shortest path the velocity at the start and end points goes to const value instantaneously meaning almost infinite acceleration. That's why we see a linear graph b/w joint angles vs time. On the other hand the slope of the curve is flat at the start and end, indicating that the velocity starts at zero, smoothly increases to a peak, and smoothly decelerates back to zero. Joint angles changing with a cubic function of time easily handle the given constraints(of 0 velocity at start and end).

2.- Why smoothness is important for real robots-

Smoothness is important for real robots because a linear trajectory implies infinite acceleration at the start and stop, which requires infinite force. In reality, this causes high 'Jerk' (the rate of change of acceleration), which creates damaging vibrations, wears out the gearbox (mechanical stress), and causes the motors to stall or overheat. Smooth trajectories (like cubic polynomials) ensure acceleration is finite and continuous, protecting the hardware.