1. Objective

The goal of this assignment was to implement four control algorithms for a differentially driven robot to follow the predefined sequence of waypoints:

T1 = [3, 0], T2 = [6, 4], T3 = [3, 4], T4 = [3, 1], T5 = [0, 3]

Initial pose: [x(0), y(0), ϕ(0)] = [0, 0, 0°]

Implemented controllers:

- Pure Pursuit Controller
- Proportional Controller (P-Controller)
- Stanley Controller
- Model Predictive Control (MPC)

The robot subscribes to the /odom topic, publishes velocity commands to /cmd_vel, stops upon reaching the final waypoint, and the actual trajectory is visualized and compared with the desired one.

2. Implemented Controllers – Summary

Controller	Core Principle	Final Key Parameters	Main Features
Pure Pursuit	Follows a lookahead point at distance L ahead of the current goal	L = 0.5 m, v = 0.5 m/s, threshold = 0.1 m	Simple, constant speed
P-Controller	Proportional control based on distance and heading error	kp_lin = 1.2, kp_ang = 2.5, threshold = 0.1 m	Easiest to implement, prone to oscillation
Stanley	Heading error + cross-track error (CTE) with front-axle projection	k = 0.8, v_max = 0.35 m/s, L = 0.3 m, threshold = 0.25 m	Dynamic speed, very robust to turns
MPC	Grid-search optimization over a 12-step prediction horizon	N = 12, dt = 0.8 s, v_ref = 0.32 m/s, threshold = 0.22 m	Multi-objective cost, computationally heavier

3. Challenges and Solutions

Ros things

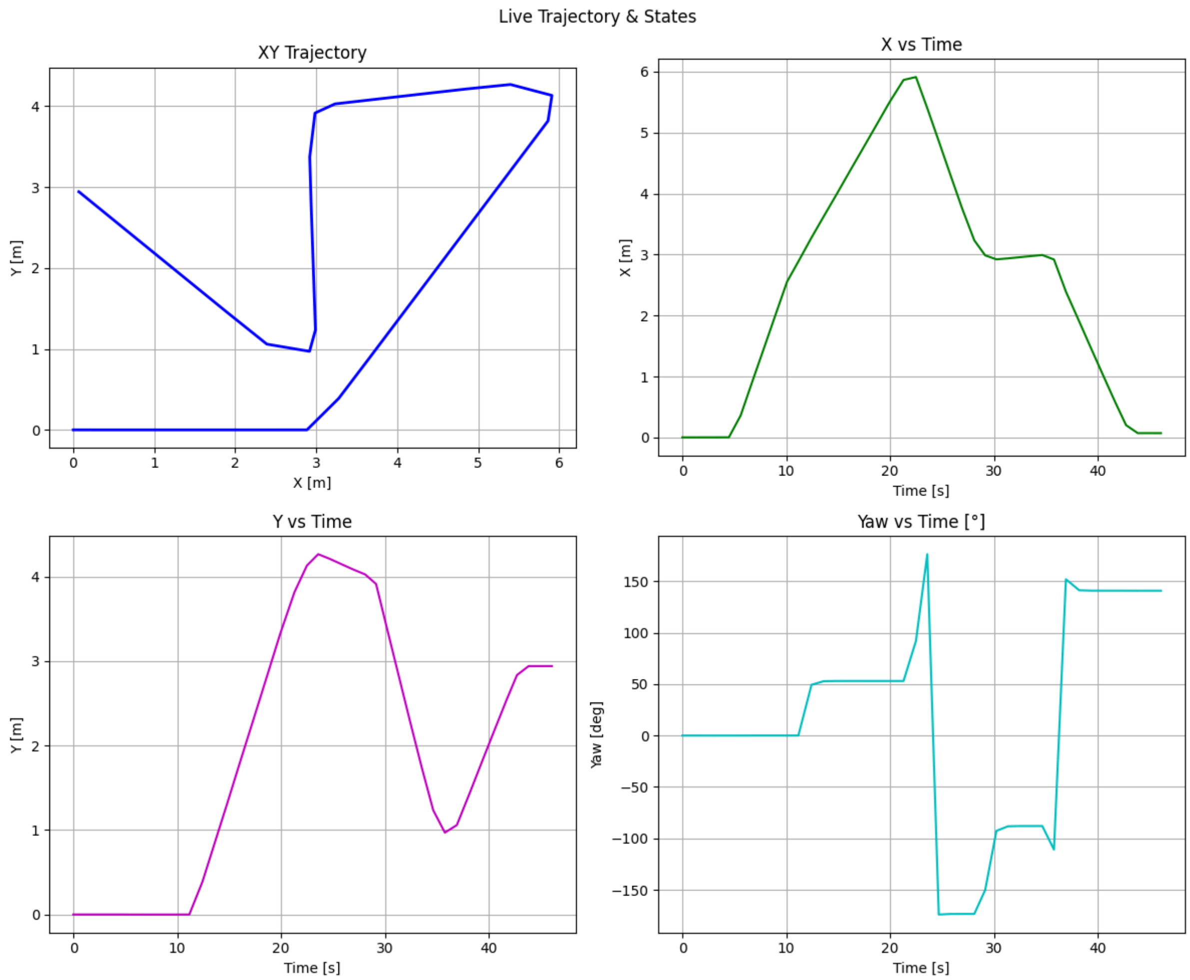
- **Missing tf_transformations package** Solution: Implemented manual quaternion-to-yaw conversion using atan2 formula, problem was in incompatible version of libraries
- **Incorect installation of setuptools** Solution: delete it and install version 58.2.0

Trajectory following algorithms:

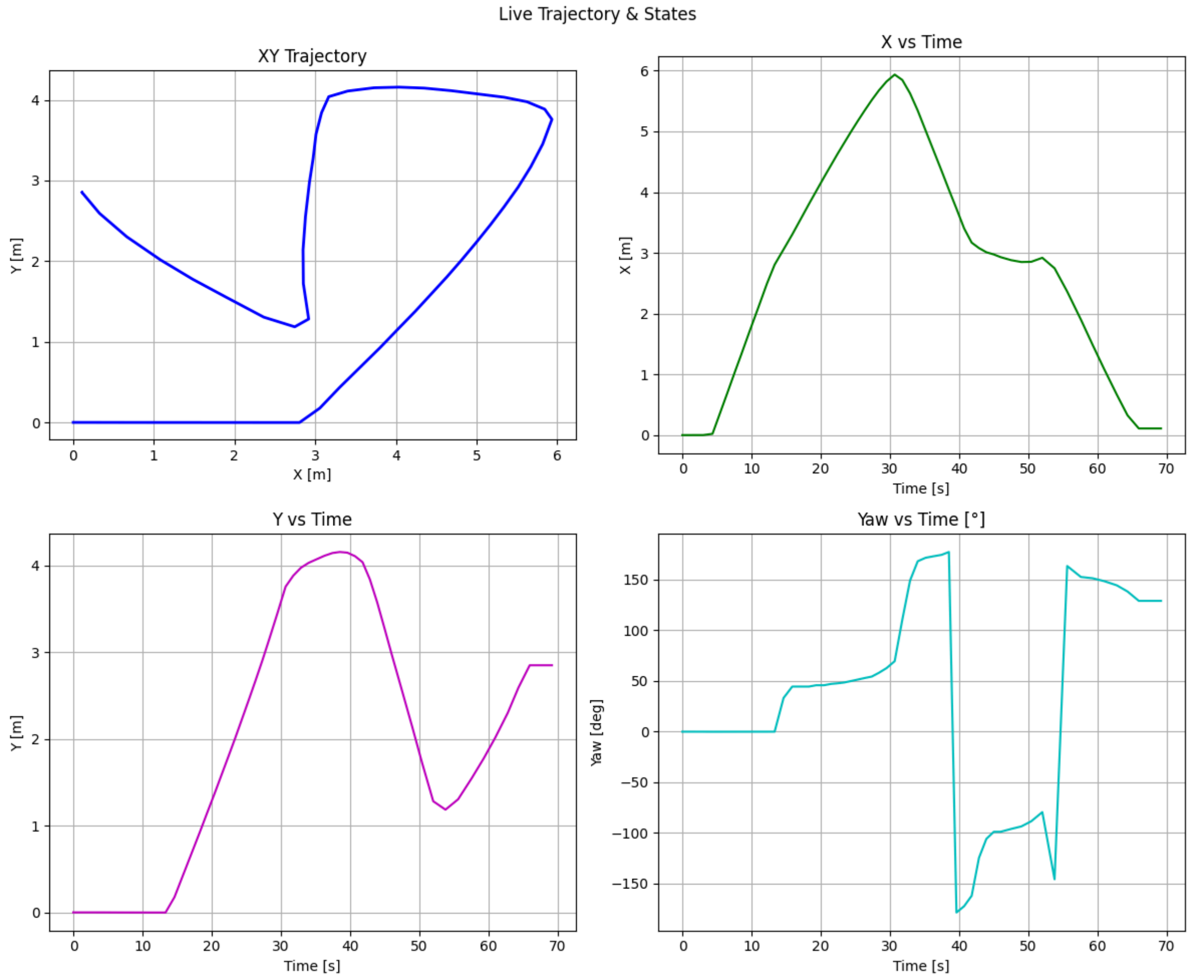
- **Stanley and MPC initially oscillated / spun around waypoints** Cause: high heading error combined with constant linear velocity led to overshooting and instability. Solution: – Introduced dynamic linear speed (reduced when heading error is large) – Correct signed cross-track error calculation in Stanley (front-axle projection) – Added heading penalty and angular rate smoothing in MPC cost function. Increased waypoint acceptance threshold to 0.22–0.25 m

4. Recorded Trajectories

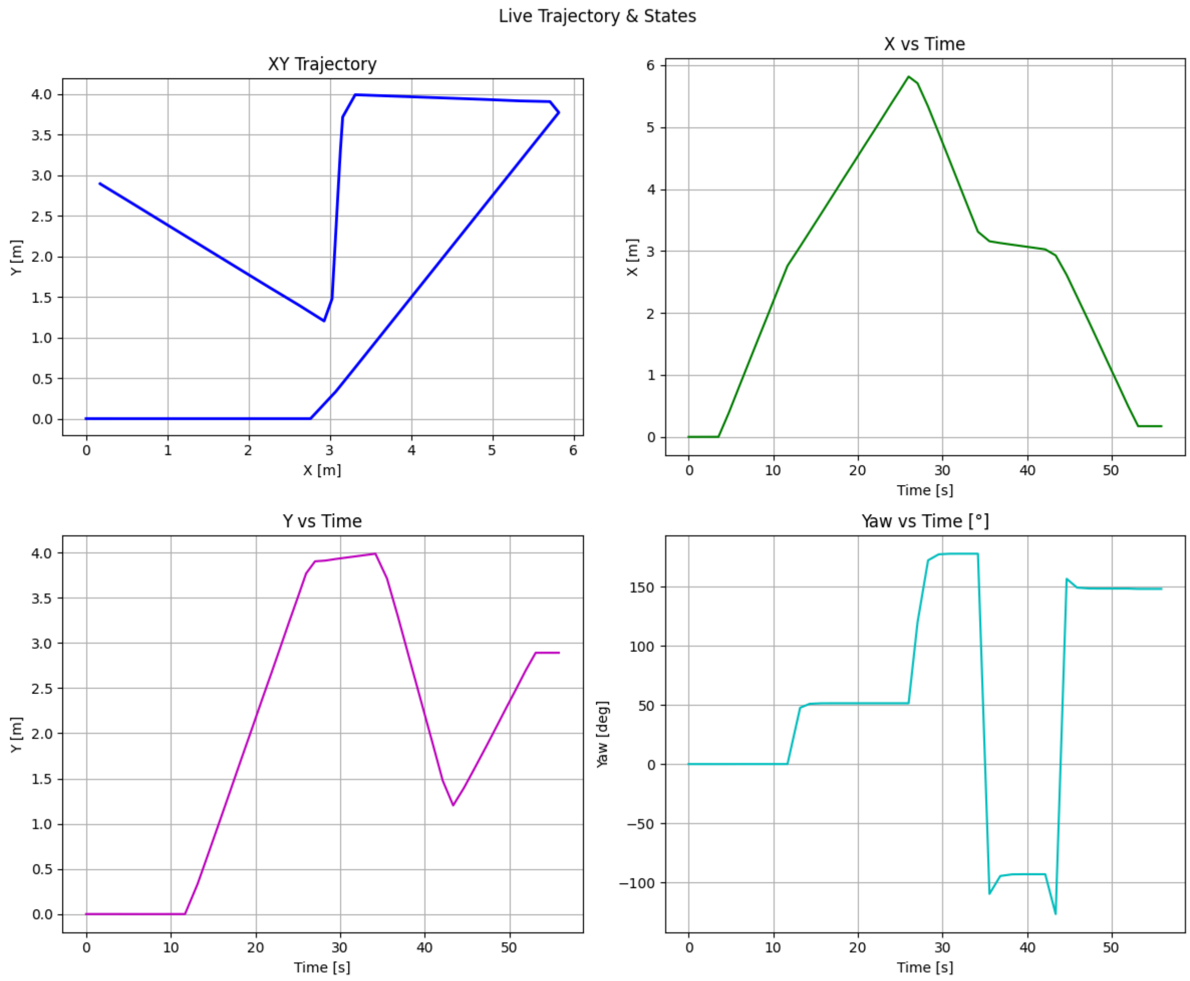
P-controller:



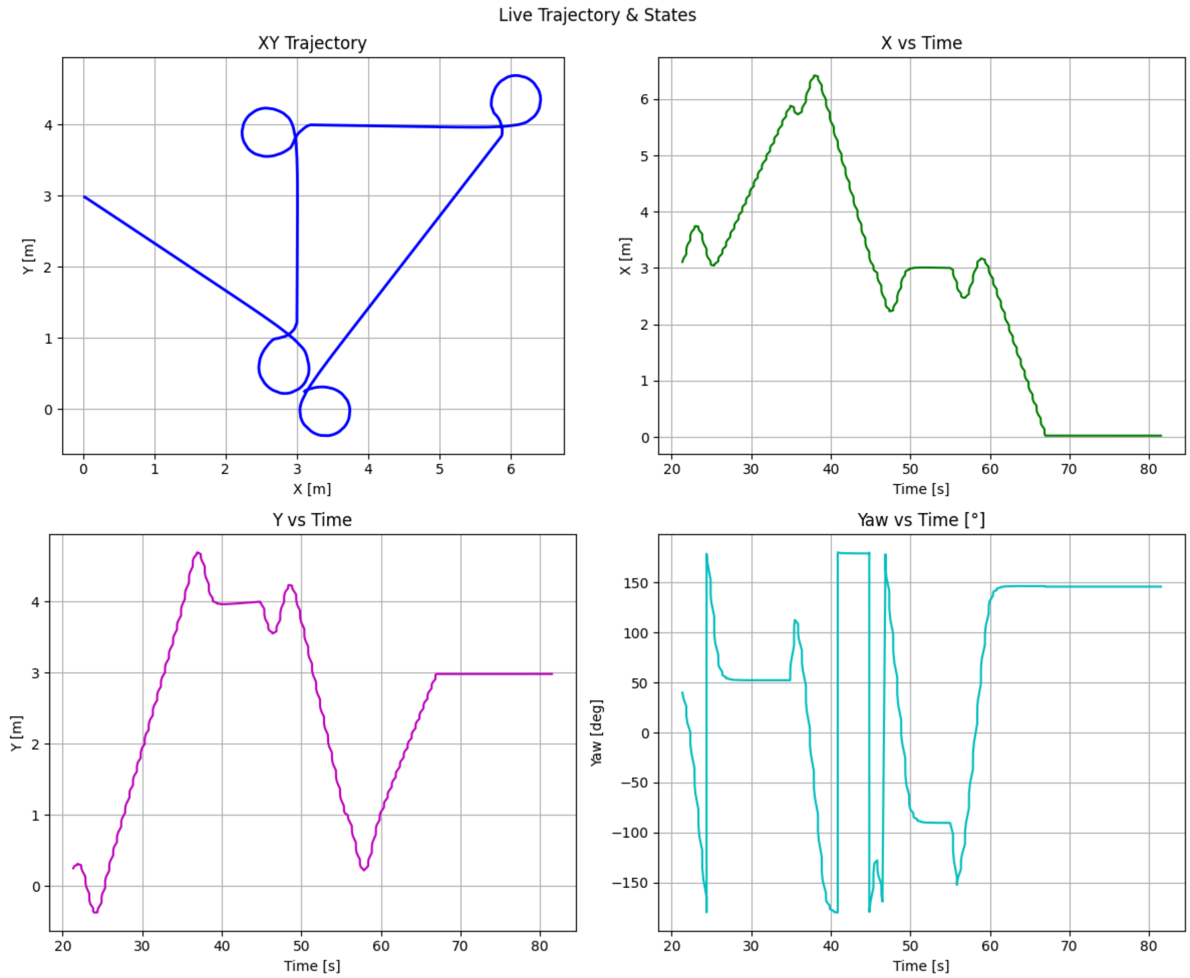
Mpc controller:



Stanley controller:



Pure Persuit:



I am not sure why the robot is spinning around each of the points. For the rest, algorithm is doing great with the *trajectory point accepted threshold* = 0.1.