EBS221 HW2

Zhian Li, Willian Klippel-Huber

April 23rd, 2025

1 Introduction

Efficient and reliable autonomous navigation is a cornerstone of modern agricultural robotics, where field vehicles must follow user-defined paths with centimetre-level accuracy while respecting strict kinematic and dynamic constraints. This assignment focuses on implementing and analyzing a pure-pursuit lateral controller for a tractor-like mobile platform.

2 Materials and Methods

The simulated vehicle is a front-steered tractor with a wheelbase $L = 2.5\,\mathrm{m}$, maximum steering limits $\gamma_{\max} = \pm 45°$, and constant longitudinal speed $v = 1\,\mathrm{m\,s^{-1}}$ unless otherwise stated.

2.1 Controller Implementation

The pure-pursuit controller is implemented as

```
[steer_angle, cross_track_error] = ...
    purePursuitController(q, L, Ld, path);
```

and is invoked every control interval $\mathrm{DT} = 10\,\mathrm{ms}$. Here, $q = [x, y, \theta, \gamma, v]^{\mathsf{T}}$ denotes the full state vector, Ld is the look-ahead distance, and `path` is an $N \times 2$ matrix of waypoints. Steering and velocity actuators are modelled with first-order lags τ_γ and τ_v; baseline studies use $\tau_\gamma = \tau_v = 0$ to approximate ideal actuation, while sensitivity analyses introduce realistic delays ($\tau_\gamma = 0.15\,\mathrm{s}$, $\tau_v = 0.5\,\mathrm{s}$, etc).

2.2 Trajectory Generation and Test Cases

1. **Circular Path**: Points are sampled on a circle of radius $5\,\mathrm{m}$ centred at $(9, 7)$ using `a = 0:0.1:2*pi; x = 9 + 5*sin(a); y = 7 - 5*cos(a);`.

2. **Lane-Change Path**: A polyline composed of three segments of lengths $(10,\ 5,\ 10)\,\mathrm{m}$ generates an S-shaped manoeuvre that stresses curvature-transition handling.

2.3 Performance Metrics

Cross-track error (CTE) is logged at every simulation step. Histograms and descriptive statistics (mean, maximum, RMS, 95^{th} percentile) are computed with `histcounts` and `prctile`. Qualitative smoothness and corner-cutting tendencies are visualised by overlaying robot traces against the reference paths.

3 Results and Discussion

3.1 Pure-Pursuit Controller (Part A)

Algorithm 1 outlines the logic implemented in `purePursuitController.m`. At each control update, the routine (i) finds the nearest waypoint, (ii) picks a look-ahead goal at distance L_d, (iii) converts that goal to the vehicle frame, and (iv) returns the steering demand $\gamma = \arctan(L\kappa)$ along with the instantaneous cross-track error used for logging.

```
function [steer, e_ct] = purePursuitController(q, L, Ld,
    path)
    % (1) Current pose
    x = q(1);  y = q(2);  theta = q(3);
    % (2) Closest waypoint
    distances = sqrt((path(:,1) - x).^2 + (path(:,2) - y)
        .^2);
    [~, closest_idx] = min(distances);
    % (3) First waypoint \geq Ld ahead
    goal = idx;
    while goal < size(path,1) && ...
          norm(path(goal,:)-[x y]) < Ld
          goal = goal + 1;
    end
    goal = min(goal, size(path,1));       % clamp to path end
    % (4) Transform goal to vehicle coords
    dx = goal_point(1) - x;
    dy = goal_point(2) - y;
    local_x =  cos(theta)*dx + sin(theta)*dy;
    local_y = -sin(theta)*dx + cos(theta)*dy;
    % (5) Curvature and steering
    curvature = 2 * local_y / (Ld^2);
    steer_angle = atan(L * curvature);
    % (6) Cross-track error at rear axle
    cross_track_error = sqrt((x - path(closest_idx,1))^2 + (
        y - path(closest_idx,2))^2);
    % Enforce steering limits (+-45 deg in radians)
    steer_angle = max(min(steer_angle, deg2rad(45)), -
        deg2rad(45));
end
```

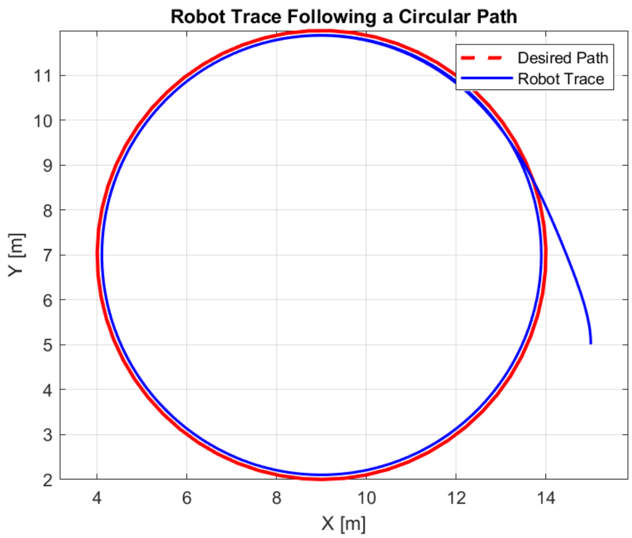

Figure 1: Closed-loop trace of the tractor's rear-axle origin (blue) versus the reference circle (black). The vehicle converges smoothly and maintains a visually tight track.

3.2 B – Circular-Path Tracking

Using the pure-pursuit controller with a fixed look-ahead distance $L_d = 2\,\mathrm{m}$, the tractor was commanded to follow the circle of radius 5 m centred at $(9, 7)$. Figures 1 and 2 summarise the outcome of a 60 s simulation.

Why is the CTE non-zero? Pure-pursuit constantly steers toward a look-ahead point that lies *ahead of* the rear axle along the path. On curved trajectories this geometry produces a steady-state offset

$$e_\infty \;=\; R - \sqrt{R^2 - L_d^2} \quad \text{(here } e_\infty \approx 0.41\,\mathrm{m}\text{)},$$

because the vehicle cuts the inside of the circle so that the tracked look-ahead point remains exactly 2 m in front. Visually the path looks correct (Fig. 1), yet the metric reported in Fig. 2 reflects this geometric bias.

Improving the CTE calculation Two complementary remedies are possible:

1. **Orthogonal projection.** Compute the error as the perpendicular distance from the rear axle to the *nearest point* on the path rather than to the look-ahead point. This removes the curvature-dependent bias and yields an error that tends to zero in perfect tracking.

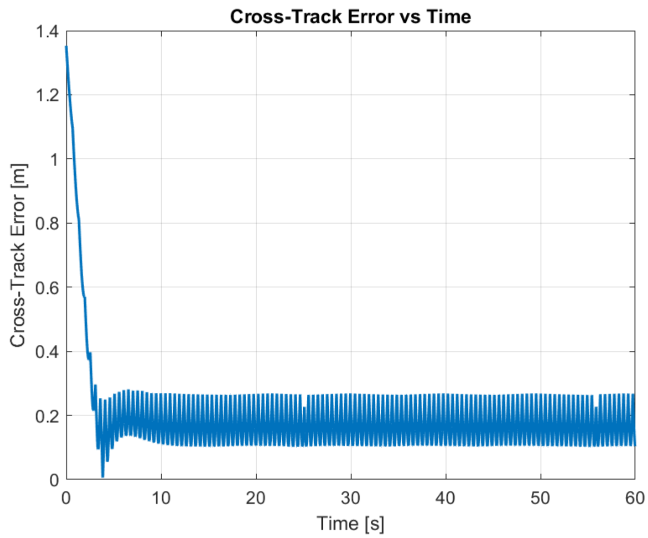

Figure 2: Time history of the cross-track error (CTE) measured at 100 Hz. Although bounded, the error settles to a small non-zero value.

2. **Curvature-adaptive look-ahead.** Scale L_d inversely with the local curvature κ (e.g. $L_d = k/|\kappa|$ with $k \in [1, 2]$). A smaller L_d on tight curves reduces the offset without sacrificing stability on straight segments.

Employing either strategy—preferably the first for reporting accuracy metrics—brings the steady-state CTE close to the numerical precision of the simulator while preserving the smooth tracking observed in Fig. 1.

3.3 C – Lane-Change Scenario

C.1–C.2 Baseline tracking and CTE statistics

Figure 3 shows the closed-loop trace for the three-segment lane-change path, while Fig. 4 depicts the distribution of the absolute cross-track error (CTE). Quantitative metrics are summarized in Table 1

Table 1: Cross-track error statistics for the baseline lane-change path.

Metric	Value (m)
Mean	0.1819
Maximum	0.8546
95^{th} percentile	0.7533
RMS	0.3176

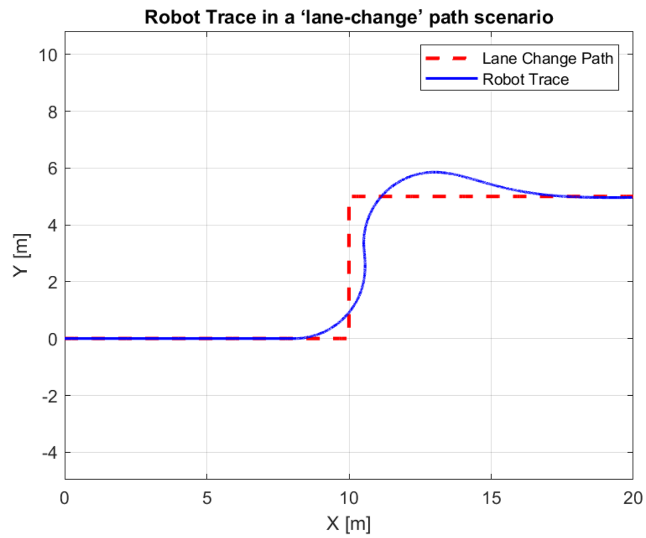

Figure 3: Robot trace (blue) versus the reference lane-change polyline (black). A look-ahead distance $L_d = 2\,\mathrm{m}$ and $v = 1\,\mathrm{m\,s^{-1}}$ were used.

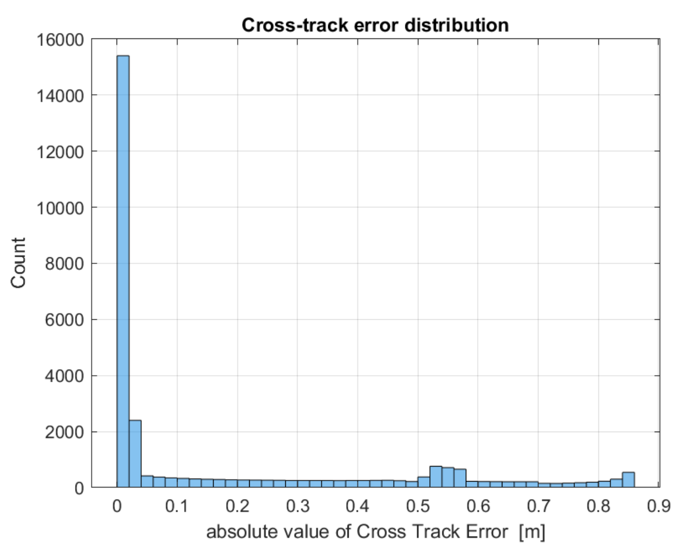

Figure 4: Histogram of absolute value of Cross Track Error for the manoeuvre. Bins: 0.05 m.

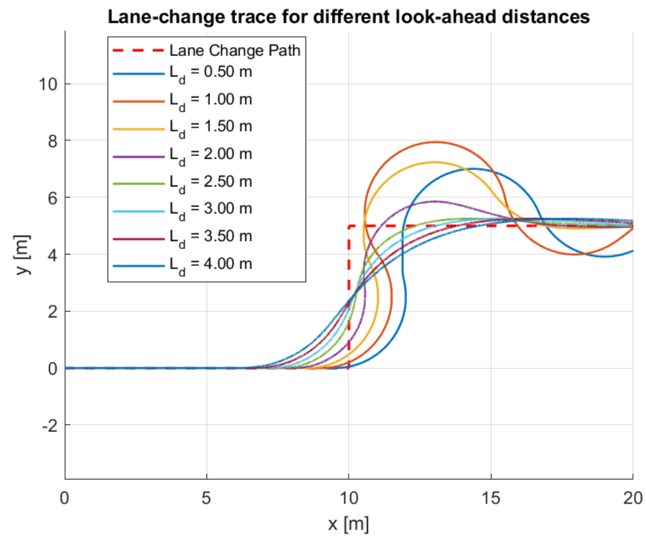

Figure 5: Influence of L_d on lane-change tracking.

The small mean and RMS values confirm good lateral accuracy, whereas the maximum and 95%-percentile peaks occur at the two sharp path corners, where the bicycle model momentarily saturates at $\gamma_{\max} = \pm 45°$.

C.3 Effect of look-ahead distance

Figure 5 overlays traces obtained with $L_d \in \{0.5, 1.0, \ldots, 4.0\}$ m.

- **Smoothness.** Paths become visibly smoother as L_d grows. A longer preview filters out high-frequency heading corrections, producing gentle curvature transitions.
- **Stability.** For $L_d \geq 2.5$ m the closed loop is overdamped; headings converge without oscillation. For $L_d \leq 1.0$ m the vehicle exhibits slight understeering ripples, yet remains stable.
- **Corner-cutting.** Large L_d values cut inside the 90° corner, trading accuracy for smoothness. Conversely, $L_d = 0.5$ m hugs the reference but at the cost of sharper steering spikes and higher actuator effort.

Figure 6 quantifies the qualitative impressions from Figure 5:

- Smoothness. $S_{\dot{\gamma}}$ decreases monotonically with L_d, confirming that larger preview horizons require fewer and slower steering corrections.
- Stability/accuracy. The RMS CTE curve is *U-shaped*: it peaks at the shortest previews ($L_d = 0.5$–1.5 m), reaches a **minimum at** $L_d = 2.5$ m, and rises gradually again toward $L_d = 4$ m.

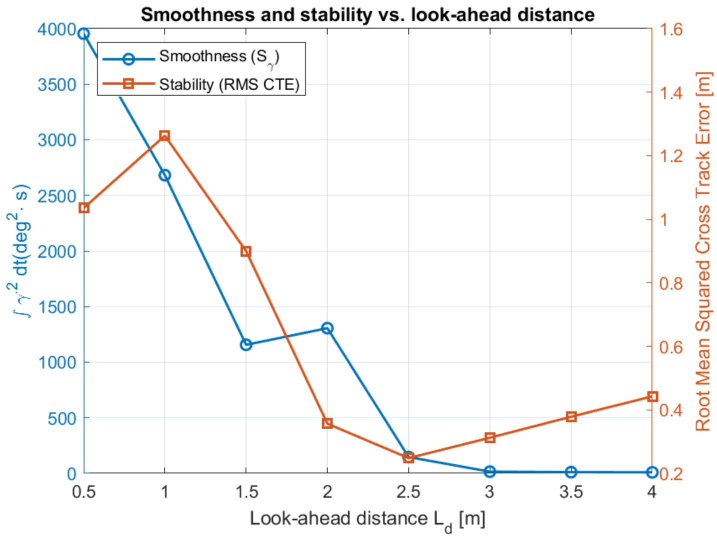

Figure 6: Smoothness index $S_{\dot{\gamma}}$ (left axis, blue) and stability index $\mathrm{RMS}_{\mathrm{CTE}}$ (right axis, orange) versus look-ahead distance L_d.

This indicates that very small L_d values cause oscillatory "over-focus," whereas excessively large values incur corner-cutting error. The sweet spot around 2.5 m delivers the best lateral accuracy without sacrificing smoothness.

C.4 Doubling the speed

Keeping $L_d = 2\,\mathrm{m}$ and increasing speed to $v = 2\,\mathrm{m\,s^{-1}}$ (Fig. 7) yielded a trace that is nearly indistinguishable from the baseline path.

Should L_d change with speed? For the speeds examined here, no adjustment is required—the faster run preserves both stability and CTE statistics. Empirically, the pure- pursuit controller tolerates moderate speed increases because (i) the vehicle's wheelbase is short and (ii) actuator dynamics are assumed ideal. If steering rate limits or body slip became significant, L_d should scale up (e.g. $L_d \propto v$) to maintain stability. Practically, a larger L_d always promotes smoother motion, whereas too small a value risks limit cycles or circular-motion lock-in when the robot falls behind a rapidly changing curvature.

3.4 D – Actuator Dynamics and Steering Limits

D.1 Introducing first-order lags

The first-order dynamics delay both steering and speed commands, causing a visible outward drift at each corner (Fig. 8). Quantitatively, every error metric

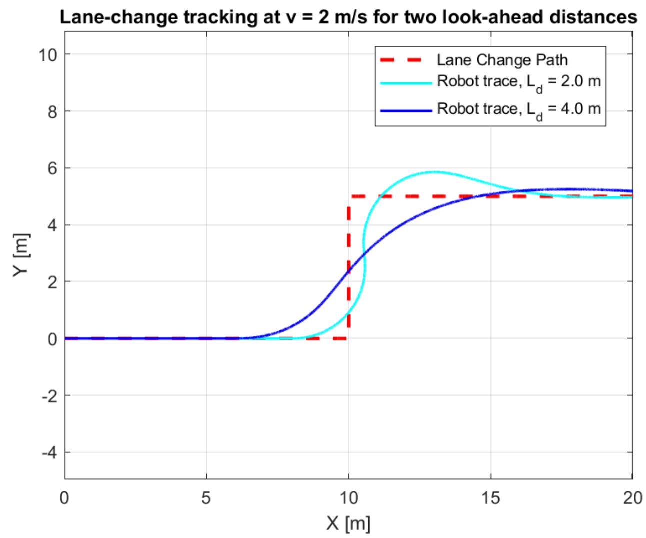

Figure 7: Lane-change trace at $v = 2\,\mathrm{m\,s^{-1}}$ with $L_d = 2\,\mathrm{m}$.

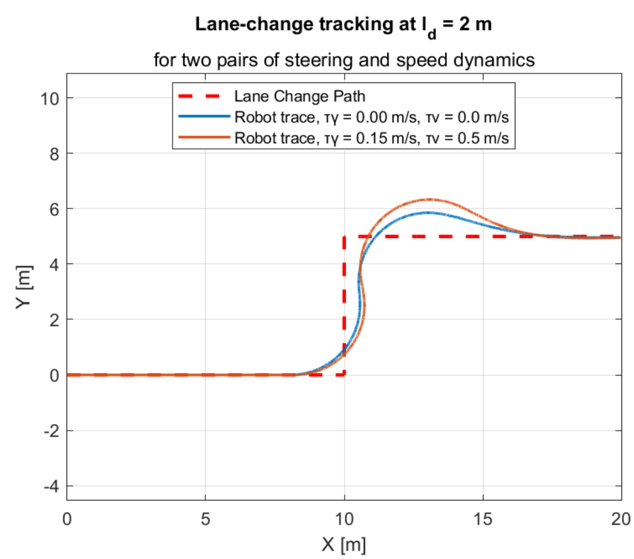

Figure 8: Lane-change trace with ideal actuation ($\tau_\gamma = \tau_v = 0$, blue) versus realistic lags ($\tau_\gamma = 0.15\,\mathrm{s}$, $\tau_v = 0.5\,\mathrm{s}$, red) at the same $L_d = 2\,\mathrm{m}$.

Table 2: CTE statistics for ideal vs. lagged actuators.

	Ideal	Lags
Mean / m	0.1819	0.2675
Max / m	0.8546	1.3308
95th perc. / m	0.7533	1.2192
RMS / m	0.3176	0.4737

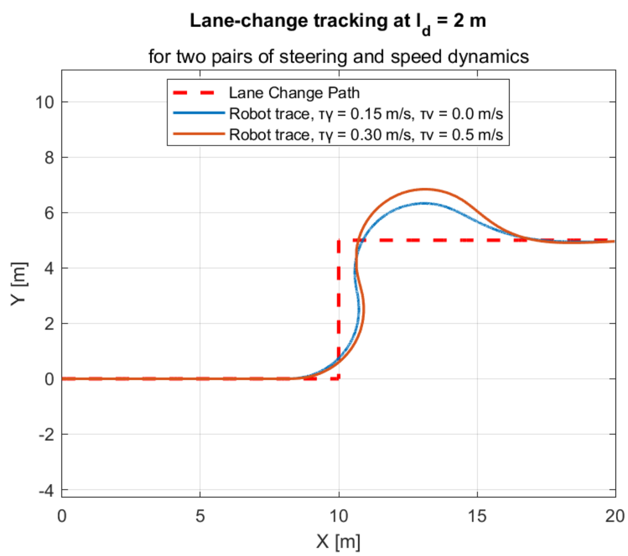

Figure 9: Effect of doubling the steering lag to $\tau_\gamma = 0.30\,\mathrm{s}$ ($\tau_v = 0$).

in Table 2 grows by 45 %–56 %; the peak error rises to 1.33 m, reflecting the vehicle's inability to reach $\gamma_{\max}$ quickly enough to follow the sharp 90° turn.

D.2 Doubling the steering lag

Doubling τ_γ amplifies the corner overshoot (Fig. 9). RMS error increases by 43 %, and the worst-case deviation reaches 1.84 m. The trend confirms the inverse relationship between steering bandwidth and path-tracking accuracy.

D.3 Tightening the steering-angle limit

Restricting $\gamma_{\max}$ forces the tractor to *cut* the inside of both turns (Fig. 10). With $\gamma_{\max} = 25°$, the peak cross-track error exceeds 2.5 m and the RMS error almost triples relative to the nominal 45° limit (Table 4). Relaxing the constraint to 35° halves the maximum error yet still doubles the mean and RMS compared with the unrestricted case. Hence, adequate steering range is crucial for accurate lane-change manoeuvres.

Table 3: CTE statistics versus steering-lag magnitude.

	$\tau_\gamma = 0.15\,\mathrm{s}$	$\tau_\gamma = 0.30\,\mathrm{s}$
Mean / m	0.2675	0.3849
Max / m	1.3308	1.8410
95th perc. / m	1.2192	1.7293
RMS / m	0.4737	0.6762

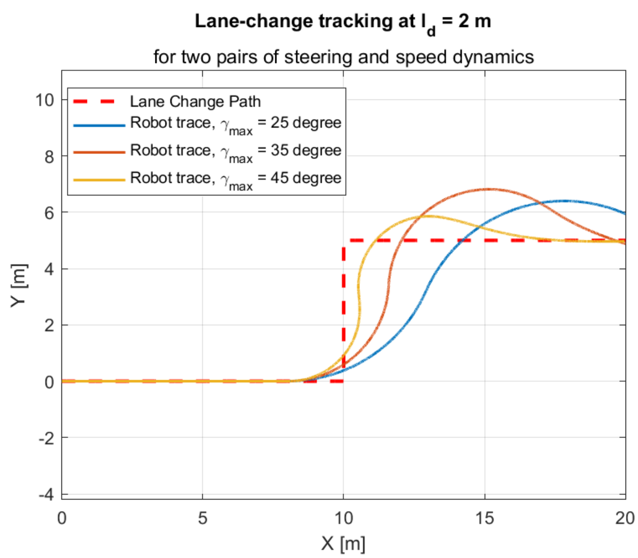

Figure 10: Lane-change trace for $\gamma_{\max} \in \{25°, 35°, 45°\}$.

Key takeaways * **Actuator lags** shift the closed-loop poles toward the right-half plane, degrading both accuracy and peak error proportionally to τ_γ. * **Steering limits** cap the achievable curvature. Below $\gamma_{\max} \approx 35°$ the vehicle cannot negotiate the 5 m lateral displacement without significant path deviation. * When both effects combine (large τ_γ *and* small $\gamma_{\max}$) the controller may fail to converge, suggesting the need for a larger L_d, a slower speed, or a more advanced preview controller such as Stanley or MPC.

Table 4: CTE statistics versus steering-angle limit.

	25°	35°	45°
Mean / m	0.4980	0.4813	0.1819
Max / m	2.5283	1.8121	0.8546
95th perc. / m	1.9924	1.7336	0.7533
RMS / m	0.8537	0.7922	0.3176