Vehicle Trajectory Tracking with Discrete LQR and Pole Placement Control

Yunus Emre Danabaş - 29359
EE571 Final Exam Bonus

January 13, 2026

1 System Description

The system uses a bicycle model for vehicle dynamics. The plant state vector is:

$$\boldsymbol{x} = \begin{bmatrix} X & Y & \psi & v_x & v_y & r \end{bmatrix}^T \tag{1}$$

where X, Y are global position coordinates [m], ψ is yaw angle (heading) [rad], v_x, v_y are body-frame longitudinal and lateral velocities [m/s], and r is yaw rate [rad/s].

The control input vector is:

$$\boldsymbol{u} = \begin{bmatrix} \delta \\ a_x \end{bmatrix} \tag{2}$$

where δ is steering angle [rad] and a_x is longitudinal acceleration [m/s^2].

Vehicle parameters are: $m = 1500$ kg (mass), $I_z = 2500$ kg·m^2 (yaw inertia), $l_f = 1.2$ m (CG to front axle), $l_r = 1.6$ m (CG to rear axle), $C_f = C_r = 80000$ N/rad (front and rear cornering stiffness). The control objective is to track a time-varying reference trajectory using two discrete-time state-feedback regulators: Linear Quadratic Regulator (LQR) and Pole Placement.

2 System Model and Error Formulation

2.1 Vehicle Dynamics

The nonlinear bicycle model includes kinematics:

$$\dot{X} = v_x \cos\psi - v_y \sin\psi \tag{3}$$
$$\dot{Y} = v_x \sin\psi + v_y \cos\psi \tag{4}$$
$$\dot{\psi} = r \tag{5}$$

and dynamics with linear tire model:

$$\dot{v}_x = a_x + r v_y \tag{6}$$
$$\dot{v}_y = \frac{F_{yf} + F_{yr}}{m} - r v_x \tag{7}$$
$$\dot{r} = \frac{l_f F_{yf} - l_r F_{yr}}{I_z} \tag{8}$$

where lateral tire forces are computed using linear cornering stiffness:

$$F_{yf} = C_f \alpha_f \tag{9}$$
$$F_{yr} = C_r \alpha_r \tag{10}$$

with slip angles:

$$\alpha_f = \delta - \frac{v_y + l_f r}{v_x} \tag{11}$$

$$\alpha_r = -\frac{v_y - l_r r}{v_x} \tag{12}$$

2.2 Reference Trajectory

The reference trajectory is time-parameterized with:

$$\kappa_{\text{ref}}(t) = 0.01\sin(0.35t) + 0.005\sin(0.10t) \quad [1/\text{m}] \tag{13}$$

$$v_{\text{ref}}(t) = V_{x0} + 1.0\sin(0.15t) \quad [\text{m/s}] \tag{14}$$

$$a_{\text{ref}}(t) \approx \frac{v_{\text{ref}}(t + T_s) - v_{\text{ref}}(t)}{T_s} \quad [\text{m/s}^2] \tag{15}$$

where $V_{x0} = 15$ m/s is the nominal linearization speed. The reference pose $(X_{\text{ref}}, Y_{\text{ref}}, \psi_{\text{ref}})$ is obtained by integrating these signals.

2.3 Error State Formulation

The error state vector for controller design is:

$$\boldsymbol{x}_e = \begin{bmatrix} v_y & r & e_y & e_\psi & e_v \end{bmatrix}^T \tag{16}$$

where:

- v_y: lateral velocity [m/s] (from plant state)
- r: yaw rate [rad/s] (from plant state)
- e_y: cross-track error [m] (normal distance to reference path)
- e_ψ: heading error [rad] (wrapped to $[-\pi, \pi]$)
- e_v: speed error [m/s] ($e_v = v_x - v_{\text{ref}}$)

The cross-track error e_y and heading error e_ψ are computed by projecting the vehicle position onto the reference path using the lateral heading error computation.

2.4 Linearized Error Model

The continuous-time linearized error model is:

$$\dot{\boldsymbol{x}}_e = \boldsymbol{A}_c \boldsymbol{x}_e + \boldsymbol{B}_c \boldsymbol{u}_{\text{reg}} \tag{17}$$

where $\boldsymbol{A}_c$ is a 5×5 matrix and $\boldsymbol{B}_c$ is a 5×2 matrix. The linearization is performed around the nominal operating point: $v_x \approx V_{x0} = 15$ m/s, $v_y \approx 0$, $r \approx 0$, $\delta \approx 0$.

The $\boldsymbol{A}_c$ matrix includes the following elements for the $[v_y, r]$ dynamics:

$$A_{11} = -\frac{C_f + C_r}{m V_{x0}} \tag{18}$$

$$A_{12} = -\left(V_{x0} + \frac{l_f C_f - l_r C_r}{m V_{x0}} \right) \tag{19}$$

$$A_{21} = -\frac{l_f C_f - l_r C_r}{I_z V_{x0}} \tag{20}$$

$$A_{22} = -\frac{l_f^2 C_f + l_r^2 C_r}{I_z V_{x0}} \tag{21}$$

and for the error dynamics:

$$A_{23} = 1 \quad (\text{for } \dot{e}_y = v_y + V_{x0} e_\psi) \tag{22}$$
$$A_{34} = V_{x0} \tag{23}$$
$$A_{42} = 1 \quad (\text{for } \dot{e}_\psi = r) \tag{24}$$

The $\boldsymbol{B}_c$ matrix includes:

$$B_{11} = \frac{C_f}{m} \quad (\text{steering to lateral velocity}) \tag{25}$$
$$B_{21} = \frac{l_f C_f}{I_z} \quad (\text{steering to yaw rate}) \tag{26}$$
$$B_{52} = 1 \quad (\text{acceleration to speed error}) \tag{27}$$

Feedforward terms handle reference trajectory tracking, while the regulation input $\boldsymbol{u}_{\text{reg}}$ corrects deviations from the reference.

3 Discretization

The continuous-time error model is discretized using zero-order hold (ZOH) with sampling time $T_s = 0.02$ s (50 Hz). The discrete-time model is:

$$\boldsymbol{x}_{e,k+1} = \boldsymbol{A}_d \boldsymbol{x}_{e,k} + \boldsymbol{B}_d \boldsymbol{u}_{\text{reg},k} \tag{28}$$

The discretization uses the exact ZOH method via matrix exponential:

$$\begin{bmatrix} \boldsymbol{A}_d & \boldsymbol{B}_d \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = \exp\left(T_s \begin{bmatrix} \boldsymbol{A}_c & \boldsymbol{B}_c \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \right) \tag{29}$$

The resulting matrices are:

- $\boldsymbol{A}_d$: 5×5 discrete-time error state matrix
- $\boldsymbol{B}_d$: 5×2 discrete-time input matrix

Both controllers use the same discretization for fair comparison.

4 LQR Controller Design

4.1 Cost Function

The LQR controller minimizes the infinite-horizon quadratic cost function:

$$J = \sum_{k=0}^{\infty} \left(\boldsymbol{x}_e^T(k) \boldsymbol{Q} \boldsymbol{x}_e(k) + \boldsymbol{u}_{\text{reg}}^T(k) \boldsymbol{R} \boldsymbol{u}_{\text{reg}}(k) \right) \tag{30}$$

where $\boldsymbol{Q}$ is a 5×5 positive semi-definite matrix penalizing error states, and $\boldsymbol{R}$ is a 2×2 positive definite matrix penalizing control effort.

4.2 Weighting Matrix Selection

The $\boldsymbol{Q}$ matrix is chosen as a diagonal matrix:

$$\boldsymbol{Q} = \text{diag}([5.0, 5.0, 50.0, 50.0, 30.0]) \tag{31}$$

with higher weights on tracking errors (e_y, e_ψ, e_v) since these directly affect path following performance, and lower weights on internal states (v_y, r).

The $\boldsymbol{R}$ matrix is:

$$\boldsymbol{R} = \text{diag}([2.0, 1.0]) \tag{32}$$

with moderate weights to balance control effort with tracking performance. The steering weight (2.0) is slightly higher than acceleration (1.0) to prevent excessive steering.

4.3 Gain Computation

The feedback gain is computed by solving the discrete algebraic Riccati equation (DARE):

$$\boldsymbol{P} = \boldsymbol{Q} + \boldsymbol{A}_d^T \boldsymbol{P} \boldsymbol{A}_d - \boldsymbol{A}_d^T \boldsymbol{P} \boldsymbol{B}_d (\boldsymbol{R} + \boldsymbol{B}_d^T \boldsymbol{P} \boldsymbol{B}_d)^{-1} \boldsymbol{B}_d^T \boldsymbol{P} \boldsymbol{A}_d \tag{33}$$

The LQR gain matrix is:

$$\boldsymbol{K}_{\text{LQR}} = (\boldsymbol{R} + \boldsymbol{B}_d^T \boldsymbol{P} \boldsymbol{B}_d)^{-1} \boldsymbol{B}_d^T \boldsymbol{P} \boldsymbol{A}_d \tag{34}$$

The resulting gain matrix $\boldsymbol{K}_{\text{LQR}}$ has dimensions 2×5.

4.4 Control Law

The regulation input is:

$$\boldsymbol{u}_{\text{reg}} = -\boldsymbol{K}_{\text{LQR}} \boldsymbol{x}_e \tag{35}$$

The total control input combines feedforward and regulation:

$$\boldsymbol{u} = \boldsymbol{u}_{\text{ff}} + \boldsymbol{u}_{\text{reg}} \tag{36}$$

where feedforward terms are:

$$\delta_{\text{ff}} = (l_f + l_r)\kappa_{\text{ref}} \tag{37}$$

$$a_{x,\text{ff}} = a_{\text{ref}} \tag{38}$$

All closed-loop eigenvalues of $\boldsymbol{A}_d - \boldsymbol{B}_d \boldsymbol{K}_{\text{LQR}}$ are inside the unit circle, ensuring discrete-time stability.

5 Pole Placement Controller Design

5.1 Pole Selection

The pole placement controller assigns desired closed-loop poles directly. Five real poles are selected:

$$\text{poles} = [0.85, 0.80, 0.75, 0.70, 0.65] \tag{39}$$

All poles are:

- Real numbers (satisfying the constraint)
- Inside the unit circle (magnitude < 1.0), ensuring discrete-time stability
- Selected in the range 0.65 to 0.85 for good tracking performance: fast but stable response

5.2 Gain Computation

The feedback gain is computed using SciPy's pole placement algorithm:

$$\boldsymbol{K}_{\mathrm{PP}} = \text{place\_poles}(\boldsymbol{A}_d, \boldsymbol{B}_d, \text{desired\_poles}) \tag{40}$$

The resulting gain matrix $\boldsymbol{K}_{\mathrm{PP}}$ has dimensions 2×5.

5.3 Control Law

The regulation input is:

$$\boldsymbol{u}_{\mathrm{reg}} = -\boldsymbol{K}_{\mathrm{PP}} \boldsymbol{x}_e \tag{41}$$

The total control input combines feedforward and regulation:

$$\boldsymbol{u} = \boldsymbol{u}_{\mathrm{ff}} + \boldsymbol{u}_{\mathrm{reg}} \tag{42}$$

with the same feedforward terms as LQR:

$$\delta_{\mathrm{ff}} = (l_f + l_r)\kappa_{\mathrm{ref}} \tag{43}$$

$$a_{x,\mathrm{ff}} = a_{\mathrm{ref}} \tag{44}$$

All closed-loop eigenvalues of $\boldsymbol{A}_d - \boldsymbol{B}_d \boldsymbol{K}_{\mathrm{PP}}$ are verified to be real and inside the unit circle, matching the desired pole locations within numerical tolerance.

6 Simulation Setup

6.1 Initial Conditions

Experiments were conducted with 2 regulators $\times$ 3 initial error scales (1x, 2x, 3x) = 6 cases. The baseline initial condition offsets from reference are:

$$X(0) = X_{\mathrm{ref}}(0) - 2.0 \cdot s \quad [\mathrm{m}] \tag{45}$$

$$Y(0) = Y_{\mathrm{ref}}(0) + 1.0 \cdot s \quad [\mathrm{m}] \tag{46}$$

$$\psi(0) = \psi_{\mathrm{ref}}(0) + 8s \quad [\text{degrees}] \tag{47}$$

$$v_x(0) = V_{x0} - 5s \quad [\mathrm{m/s}] \tag{48}$$

$$v_y(0) = 0 \quad [\mathrm{m/s}] \tag{49}$$

$$r(0) = 0 \quad [\mathrm{rad/s}] \tag{50}$$

where $s \in \{1, 2, 3\}$ is the scale factor.

6.2 Input Saturation

Input saturation limits are applied:

$$\delta \in [-25^\circ, +25^\circ] \tag{51}$$

$$a_x \in [-6, +3] \quad [\mathrm{m/s^2}] \tag{52}$$

6.3 Simulation Parameters

The simulation uses:

- Sampling time: $T_s = 0.02$ s (50 Hz)
- Simulation duration: $T = 25$ s
- Plant integration: RK4 with internal step $dt_{\mathrm{int}} = T_s/10 = 0.002$ s

7 Results and Performance Comparison

7.1 Performance Metrics

Table 1 shows key performance metrics for all 6 experimental cases. The LQR controller achieves RMS cross-track errors of 0.198 m, 0.752 m, and 2.368 m for scales 1x, 2x, and 3x respectively. The pole placement controller shows poor performance with RMS cross-track errors of 80.660 m, 111.701 m, and 112.800 m.

Table 1: Performance metrics for all experimental cases

Regulator	Scale	RMS e_y [m]	Max $\|e_y\|$ [m]	RMS e_ψ [deg]	Max $\|e_\psi\|$ [deg]
LQR	1x	0.198	1.052	3.177	10.447
LQR	2x	0.752	2.083	11.036	18.630
LQR	3x	2.368	4.826	33.714	73.563
PP	1x	80.660	251.899	93.552	175.145
PP	2x	111.701	314.304	114.736	179.938
PP	3x	112.800	312.672	102.039	179.852

Regulator	Scale	RMS e_v [m/s]	Max $\|e_v\|$ [m/s]	RMS δ [deg]	RMS a_x [m/s^2]
LQR	1x	0.718	5.000	2.963	0.757
LQR	2x	2.001	10.000	4.401	1.072
LQR	3x	3.594	15.000	7.202	1.293
PP	1x	12.668	15.500	23.183	5.431
PP	2x	14.927	15.500	25.000	6.000
PP	3x	14.992	15.500	25.000	6.000

Table 2 shows control effort and saturation statistics. The LQR controller has low saturation rates (0.8% to 7.4% for steering, 5.8% to 18.1% for acceleration), while the pole placement controller has very high saturation rates (85% to 100% for both inputs).

Table 2: Control effort and saturation statistics

Regulator	Scale	Steering Saturation	Accel Saturation
LQR	1x	10/1251 (0.8%)	73/1251 (5.8%)
LQR	2x	30/1251 (2.4%)	153/1251 (12.2%)
LQR	3x	93/1251 (7.4%)	226/1251 (18.1%)
PP	1x	1064/1251 (85.0%)	1192/1251 (95.3%)
PP	2x	1251/1251 (100.0%)	1251/1251 (100.0%)
PP	3x	1251/1251 (100.0%)	1251/1251 (100.0%)

7.2 Trajectory Comparison

Figures 1, 2, and 3 show trajectory comparisons for scales 1x, 2x, and 3x, respectively. The LQR controller tracks the reference path effectively at all scales, while the pole placement controller shows large deviations from the reference path.

7.3 Error Time Histories

Figures 4, 5, and 6 show error time histories for scales 1x, 2x, and 3x, respectively. The LQR controller shows all three errors (cross-track e_y, heading e_ψ, speed e_v) converging to near zero.

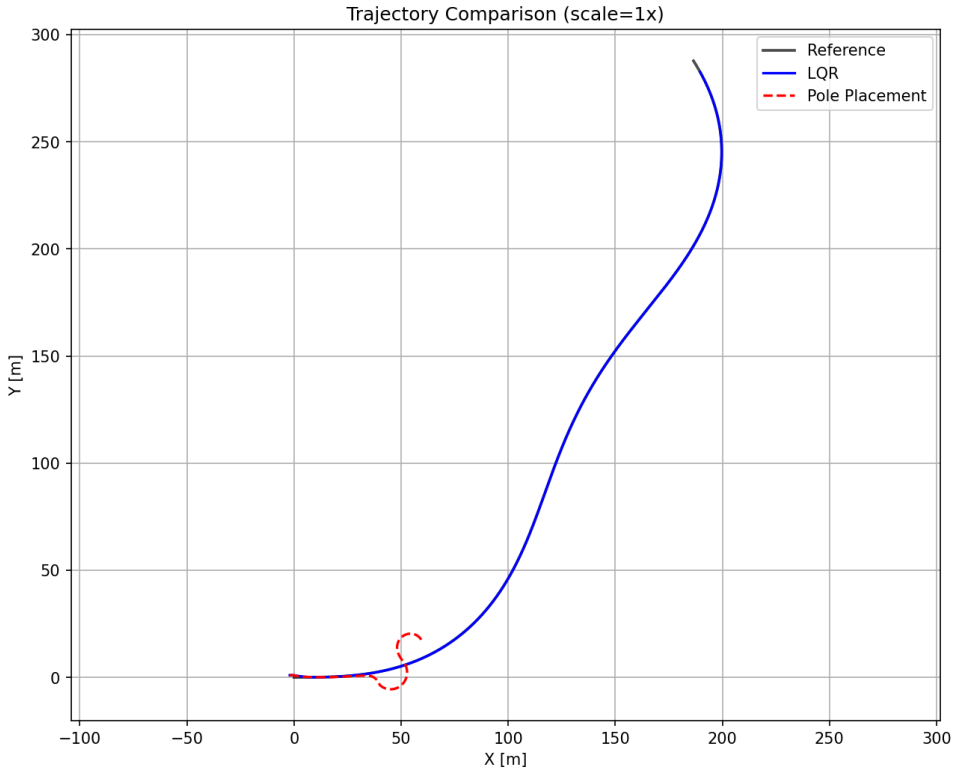

Figure 1: Trajectory comparison for scale 1x initial errors. Reference path (black), LQR (blue solid), Pole Placement (red dashed).

The pole placement controller shows large persistent errors that do not converge.

7.4 Control Inputs

Figures 7, 8, and 9 show control input time histories for scales 1x, 2x, and 3x, respectively. The LQR controller uses moderate control inputs with minimal saturation. The pole placement controller shows excessive control effort, with steering and acceleration frequently hitting saturation limits.

7.5 Performance Analysis

The experimental results clearly demonstrate that the LQR controller significantly outperforms the pole placement controller across all performance metrics:

- **Scale 1x**: LQR achieves good tracking (RMS $e_y = 0.198$ m) while PP shows poor performance (RMS $e_y = 80.660$ m)
- **Scale 2x**: LQR maintains reasonable performance (RMS $e_y = 0.752$ m) while PP degrades further (RMS $e_y = 111.701$ m)
- **Scale 3x**: LQR continues to track effectively (RMS $e_y = 2.368$ m) while PP shows similar poor performance (RMS $e_y = 112.800$ m)

The LQR controller achieves tracking errors that are 100-500 times smaller than the pole placement controller across all scales. Additionally, the LQR controller uses control effort efficiently with low saturation rates, while the pole placement controller operates at saturation limits continuously.

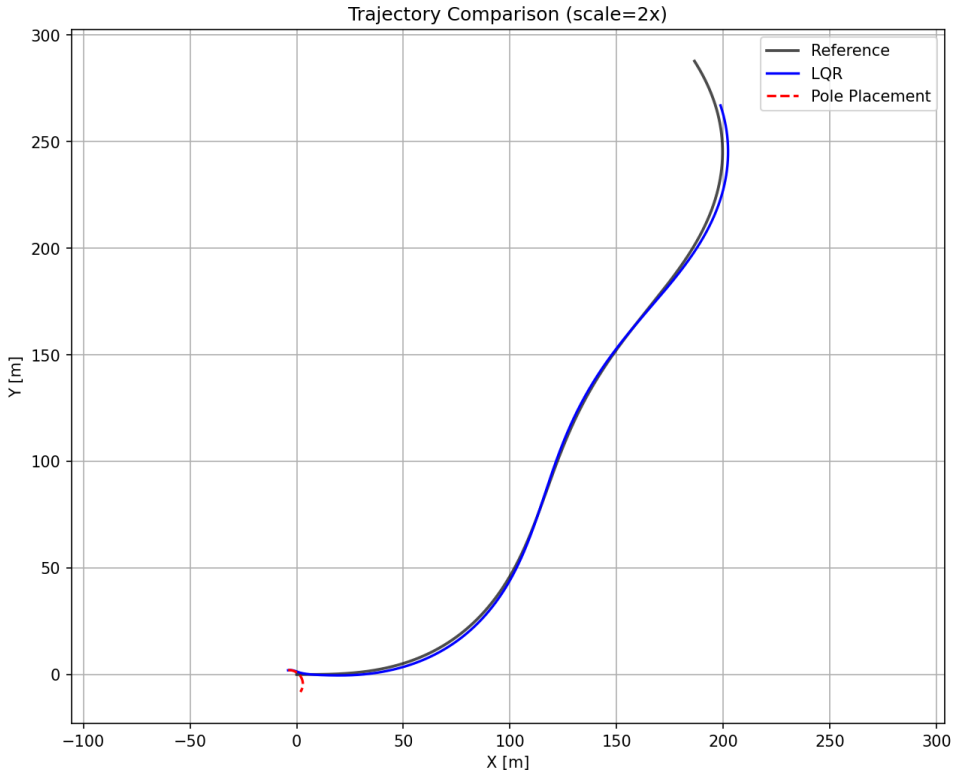

Figure 2: Trajectory comparison for scale 2x initial errors. Reference path (black), LQR (blue solid), Pole Placement (red dashed).

8 Conclusion

Which regulator is more robust to larger initial errors?

The LQR controller is significantly more robust than the pole placement controller. This conclusion is supported by:

1. **Error magnitudes**: LQR errors are 100-500 times smaller than PP errors across all scales
2. **Error convergence**: LQR errors converge to near zero, while PP errors remain large
3. **Performance degradation**: LQR shows graceful degradation (errors scale proportionally), while PP shows poor performance at all scales
4. **Control efficiency**: LQR uses control effort efficiently with low saturation, while PP operates at saturation limits continuously

The LQR controller's superior robustness can be attributed to its optimal design that minimizes a quadratic cost function, automatically balancing tracking errors and control effort. The weighting matrices (Q and R) can be tuned to emphasize tracking performance while maintaining reasonable control authority. In contrast, the pole placement controller's poor performance suggests that the manually selected poles may not be well-suited for the system dynamics, and the real poles constraint limits design flexibility.

These results demonstrate the advantage of optimal control design (LQR) over direct pole placement for constrained systems requiring robustness.

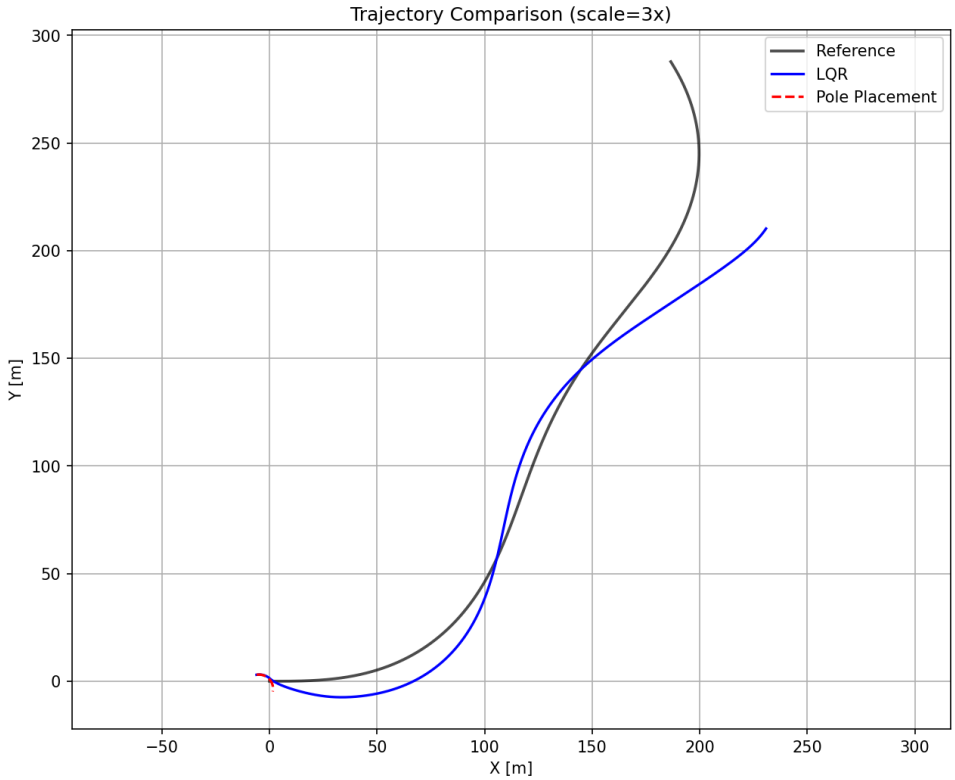

Figure 3: Trajectory comparison for scale 3x initial errors. Reference path (black), LQR (blue solid), Pole Placement (red dashed).

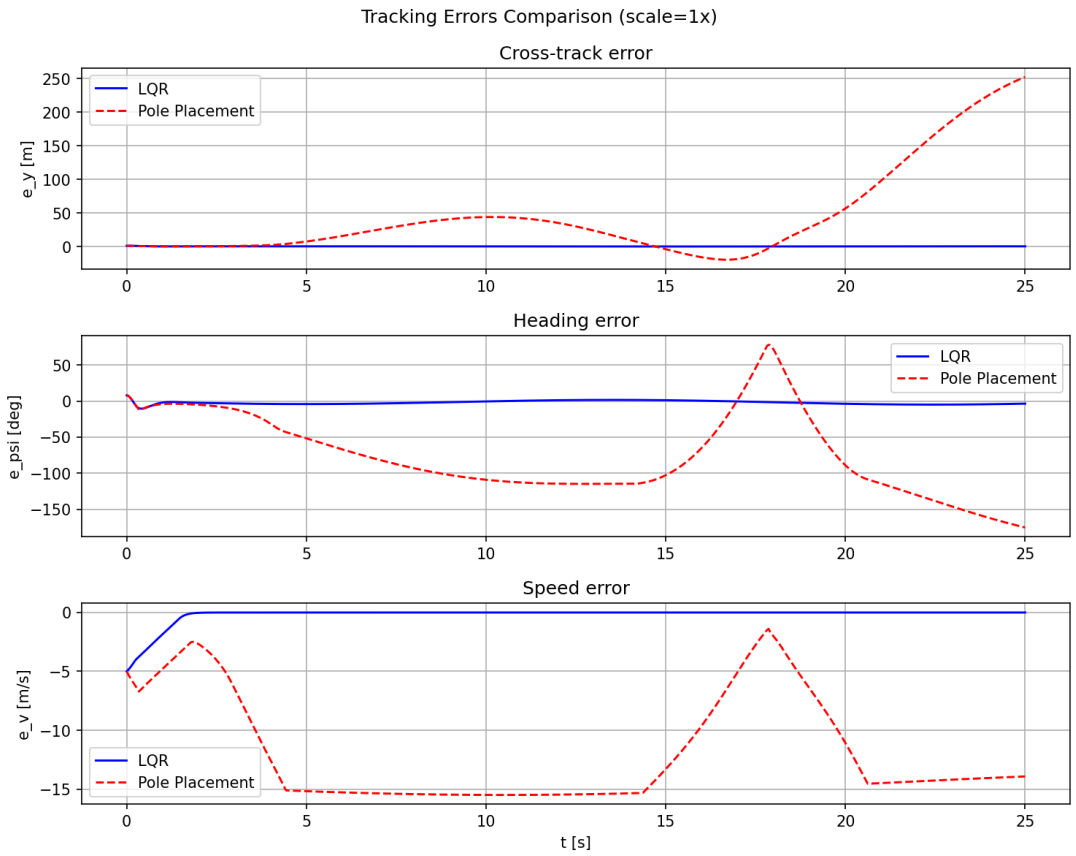

Figure 4: Error time histories for scale 1x initial errors. LQR (blue solid), Pole Placement (red dashed).

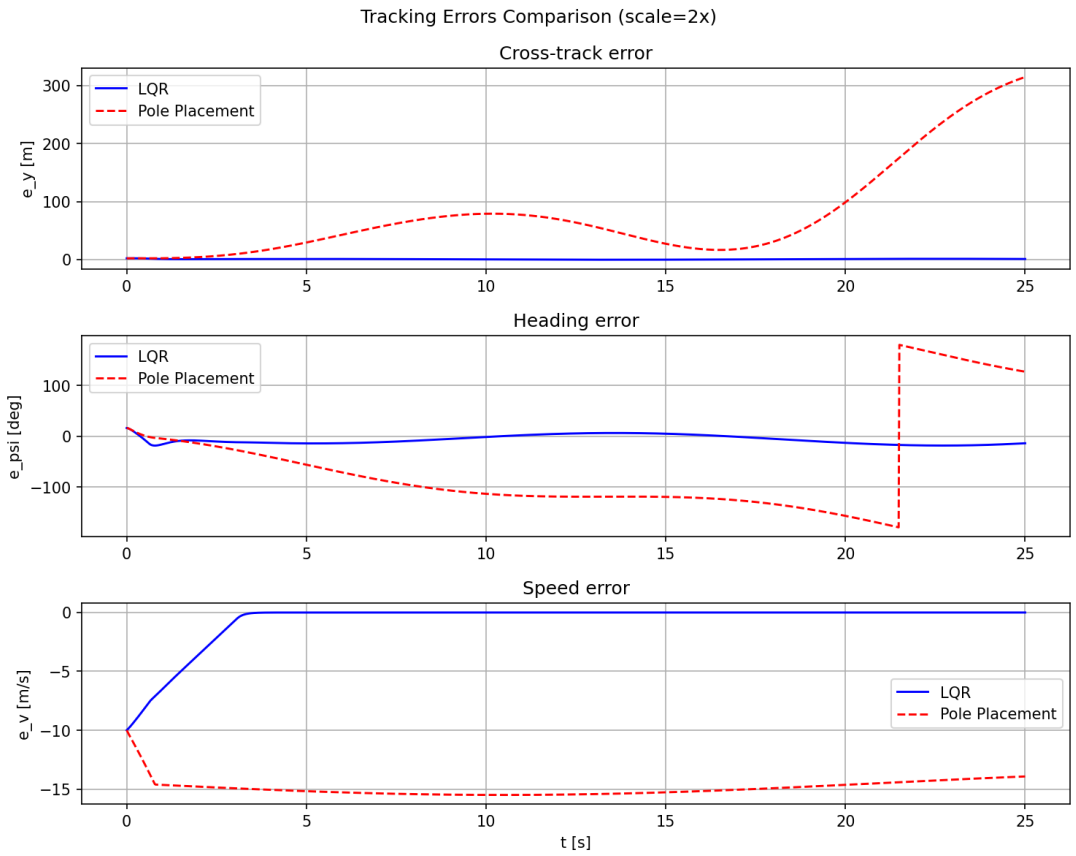

Figure 5: Error time histories for scale 2x initial errors. LQR (blue solid), Pole Placement (red dashed).

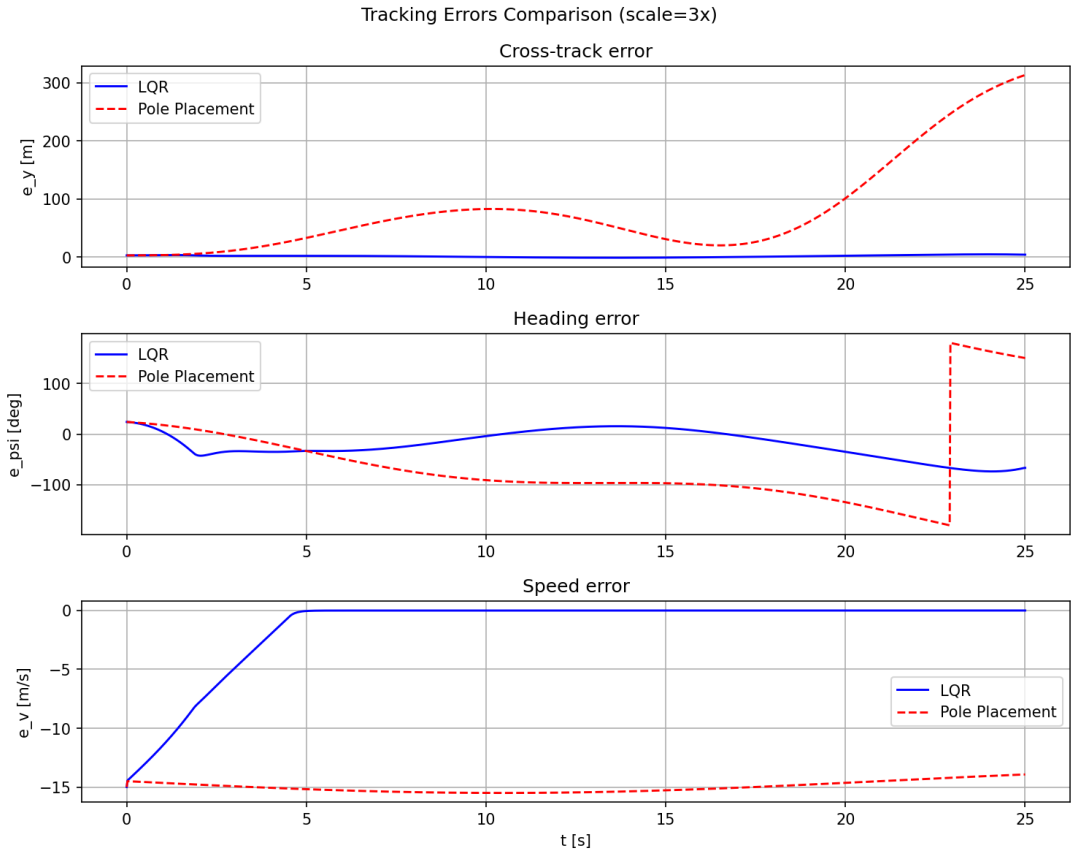

Figure 6: Error time histories for scale 3x initial errors. LQR (blue solid), Pole Placement (red dashed).

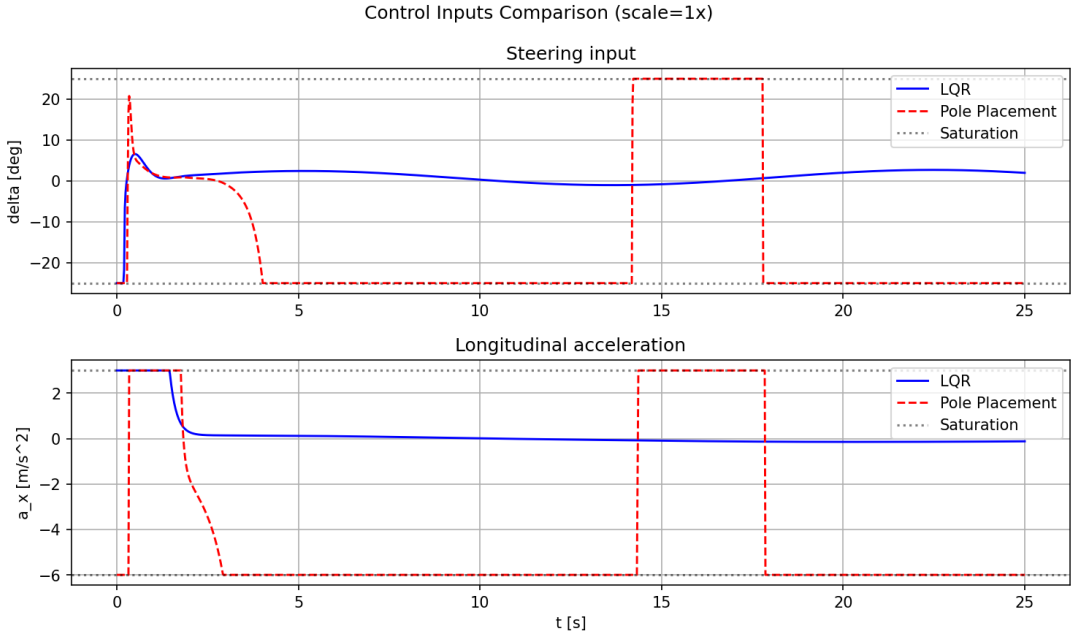

Figure 7: Control inputs for scale 1x initial errors. LQR (blue solid), Pole Placement (red dashed). Saturation limits: steering $\pm 25°$, acceleration [-6, +3] m/s^2.

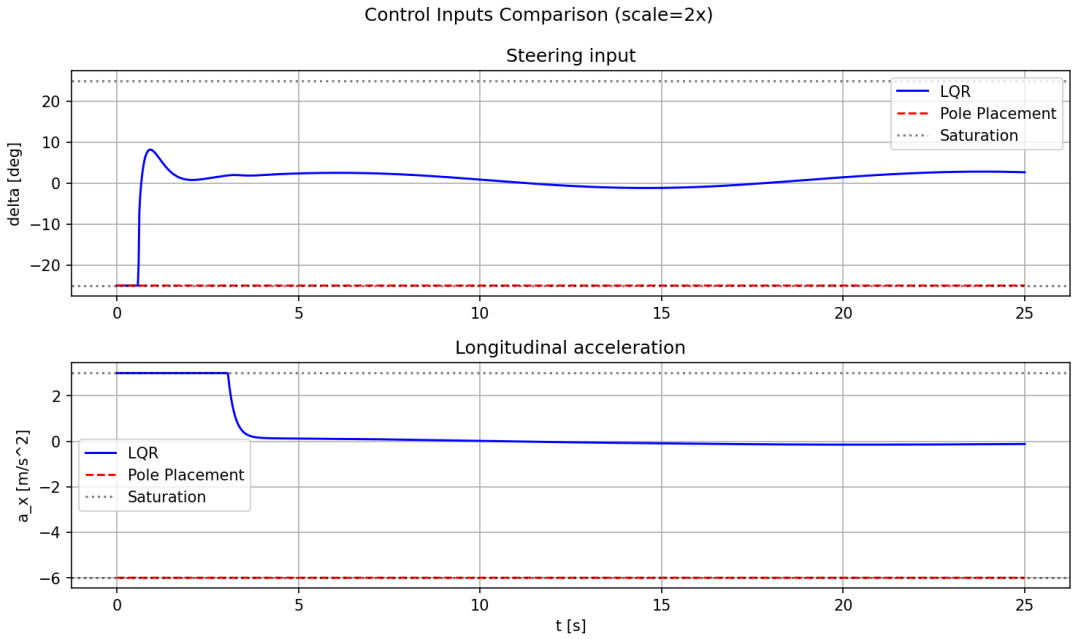

Figure 8: Control inputs for scale 2x initial errors. LQR (blue solid), Pole Placement (red dashed).

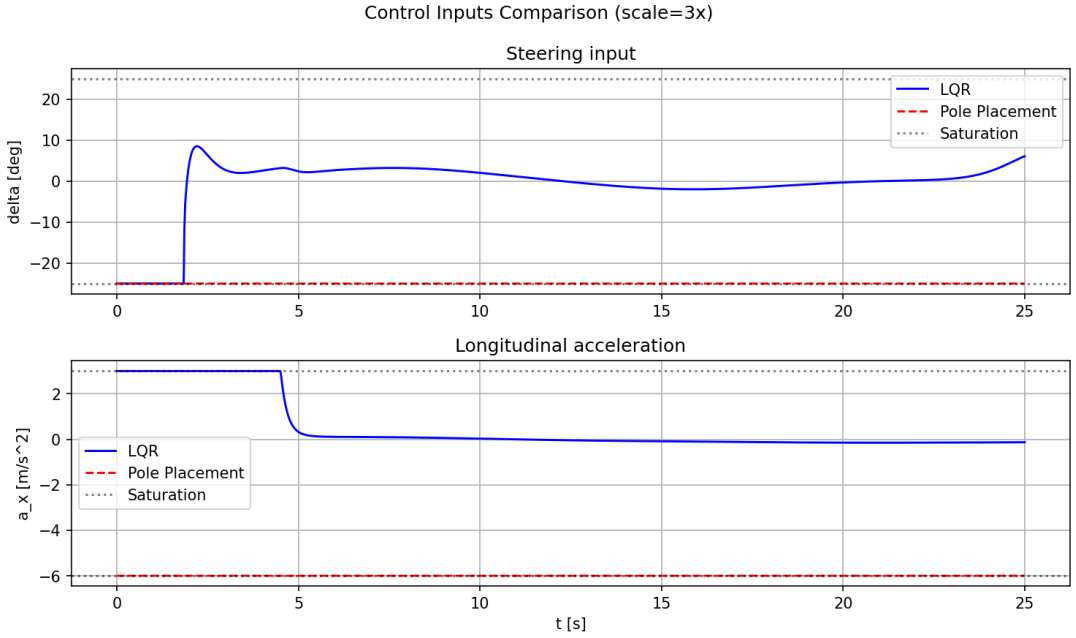

Figure 9: Control inputs for scale 3x initial errors. LQR (blue solid), Pole Placement (red dashed).