Vehicle Trajectory Tracking with Discrete LQR and Pole Placement Control

EE571 Final Exam Bonus

January 13, 2026

Abstract

This report presents a comparison of two discrete-time state-feedback regulators for vehicle trajectory tracking: a Linear Quadratic Regulator (LQR) and a Pole Placement controller. The controllers are designed using a linearized error-state model and evaluated on a nonlinear bicycle-model plant. The performance of both regulators is assessed across three different initial error scales to evaluate robustness. Experimental results show that the LQR controller demonstrates superior robustness to larger initial errors, maintaining stable tracking performance while the pole placement controller exhibits significant degradation at increased error scales. The comparison is based on quantitative metrics including RMS errors, maximum errors, and control effort, along with visual trajectory and error time history plots.

1 Introduction

1.1 Problem Statement

Vehicle trajectory tracking is a fundamental problem in autonomous vehicle control, requiring precise path following while maintaining stability and performance. This project addresses the design and comparison of two discrete-time state-feedback regulators for tracking a time-parameterized reference trajectory using a linearized error-state model for controller design, while the closed-loop system is evaluated on a nonlinear bicycle-model plant.

The control objective is to design two regulators:

1. A discrete-time infinite-horizon Linear Quadratic Regulator (LQR)
2. A discrete-time pole placement regulator with real poles only

Both controllers must stabilize the error states and track the reference trajectory while operating under input saturation constraints.

1.2 Objectives

The main objectives of this work are:

- Design two discrete-time state-feedback regulators (LQR and pole placement)
- Evaluate controller performance across different initial error magnitudes (1x, 2x, 3x scaling)
- Compare regulator robustness using quantitative metrics and visual analysis
- Determine which regulator handles larger initial errors more effectively

1.3 Report Structure

This report is organized as follows: Section 2 presents the system model and error formulation. Section 3 details the controller design methods for both LQR and pole placement. Section 4 presents experimental results and performance comparisons. Section 5 provides discussion and conclusions.

2 System Model and Error Formulation

2.1 Vehicle Model

The nonlinear plant uses a bicycle model with the following state vector:

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

where:

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

The control input vector is:

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

where:

- δ: steering angle [rad]
- a_x: longitudinal acceleration [m/s^2]

The vehicle parameters used are: $m = 1500$ kg, $I_z = 2500$ kg·m^2, $l_f = 1.2$ m, $l_r = 1.6$ m, $C_f = 80000$ N/rad, $C_r = 80000$ N/rad.

The nonlinear dynamics include 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 the lateral tire forces are computed using linear cornering stiffness: $F_{yf} = C_f \alpha_f$ and $F_{yr} = C_r \alpha_r$.

2.2 Reference Trajectory

The reference trajectory is time-parameterized with the following signals:

- Curvature: $\kappa_{\mathrm{ref}}(t) = 0.01\sin(0.35t) + 0.005\sin(0.10t)$ [1/m]
- Speed: $v_{\mathrm{ref}}(t) = V_{x0} + 1.0\sin(0.15t)$ with $V_{x0} = 15$ m/s
- Acceleration: $a_{\mathrm{ref}}(t) \approx \frac{v_{\mathrm{ref}}(t+T_s)-v_{\mathrm{ref}}(t)}{T_s}$

The reference pose $(X_{\mathrm{ref}}, Y_{\mathrm{ref}}, \psi_{\mathrm{ref}})$ is obtained by integrating these signals.

2.3 Tracking Errors

The error state vector used for controller design is:

$$\boldsymbol{x}_e = \begin{bmatrix} v_y & r & e_y & e_\psi & e_v \end{bmatrix}^T \tag{9}$$

where:

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

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

2.4 Linearized Error Model

The continuous-time linearized error model is:

$$\dot{\boldsymbol{x}}_e = \boldsymbol{A}_c\boldsymbol{x}_e + \boldsymbol{B}_c\boldsymbol{u}_{\mathrm{reg}} \tag{10}$$

where:

- $\boldsymbol{A}_c$ is a 5×5 matrix representing the error state dynamics
- $\boldsymbol{B}_c$ is a 5×2 matrix representing the input-to-error-state coupling
- $\boldsymbol{u}_{\mathrm{reg}}$ is the regulation input (combined with feedforward)

The linearization is performed around the nominal operating point: $v_x \approx V_{x0} = 15$ m/s, $v_y \approx 0$, $r \approx 0$, $\delta \approx 0$.

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

3 Controller Design

3.1 Discretization

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

$$\boldsymbol{x}_{e,k+1} = \boldsymbol{A}_d \boldsymbol{x}_{e,k} + \boldsymbol{B}_d \boldsymbol{u}_{\text{reg},k} \tag{11}$$

The discretization uses the exact ZOH method via matrix exponential:

$$\begin{bmatrix} \boldsymbol{A}_d & \boldsymbol{B}_d \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} = \exp\left(T_s \begin{bmatrix} \boldsymbol{A}_c & \boldsymbol{B}_c \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \right) \tag{12}$$

The resulting matrices are:

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

The same discretization is used for both controllers to ensure fair comparison.

3.2 Regulator 1: Discrete-Time LQR

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

$$J = \sum_{k=0}^{\infty} \left(\boldsymbol{x}_e^T(k) \boldsymbol{Q} \boldsymbol{x}_e(k) + \boldsymbol{u}_{\text{reg}}^T(k) \boldsymbol{R} \boldsymbol{u}_{\text{reg}}(k) \right) \tag{13}$$

3.2.1 Weighting Matrix Design

The $\boldsymbol{Q}$ matrix (5 × 5, positive semi-definite) is chosen as a diagonal matrix:

$$\boldsymbol{Q} = \text{diag}([5.0, 5.0, 50.0, 50.0, 30.0]) \tag{14}$$

Rationale:

- Higher weights on tracking errors (e_y, e_ψ, e_v) since these directly affect path following performance
- Lower weights on internal states (v_y, r) as they are intermediate variables
- The values emphasize tracking accuracy while allowing reasonable control authority

The $\boldsymbol{R}$ matrix (2 × 2, positive definite) is:

$$\boldsymbol{R} = \text{diag}([2.0, 1.0]) \tag{15}$$

Rationale:

- Moderate weights to balance control effort with tracking performance
- Steering weight (2.0) slightly higher than acceleration (1.0) to prevent excessive steering
- Values chosen to allow sufficient control authority while penalizing excessive inputs

3.2.2 Gain Computation

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

$$\boldsymbol{P} = \boldsymbol{Q} + \boldsymbol{A}_d^T \boldsymbol{P} \boldsymbol{A}_d - \boldsymbol{A}_d^T \boldsymbol{P} \boldsymbol{B}_d (\boldsymbol{R} + \boldsymbol{B}_d^T \boldsymbol{P} \boldsymbol{B}_d)^{-1} \boldsymbol{B}_d^T \boldsymbol{P} \boldsymbol{A}_d \tag{16}$$

The LQR gain matrix is:

$$\boldsymbol{K}_{\text{LQR}} = (\boldsymbol{R} + \boldsymbol{B}_d^T \boldsymbol{P} \boldsymbol{B}_d)^{-1} \boldsymbol{B}_d^T \boldsymbol{P} \boldsymbol{A}_d \tag{17}$$

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

3.2.3 Control Law

The regulation input is:

$$\boldsymbol{u}_{\text{reg}} = -\boldsymbol{K}_{\text{LQR}} \boldsymbol{x}_e \tag{18}$$

The total control input combines feedforward and regulation:

$$\boldsymbol{u} = \boldsymbol{u}_{\text{ff}} + \boldsymbol{u}_{\text{reg}} \tag{19}$$

3.2.4 Closed-Loop Stability

All closed-loop eigenvalues of $\boldsymbol{A}_d - \boldsymbol{B}_d \boldsymbol{K}_{\text{LQR}}$ are inside the unit circle, ensuring discrete-time stability. The maximum eigenvalue magnitude is approximately 0.940, well within the stability region.

3.3 Regulator 2: Discrete-Time Pole Placement

The pole placement controller assigns desired closed-loop poles directly. The constraint is that all poles must be real (no complex poles allowed).

3.3.1 Pole Selection

The five desired poles are chosen as:

$$\text{poles} = [0.85, 0.80, 0.75, 0.70, 0.65] \tag{20}$$

Rationale:

- All poles are real numbers, satisfying the constraint
- All poles are inside the unit circle (magnitude < 1.0), ensuring discrete-time stability
- Poles selected in the range 0.65 to 0.85 for good tracking performance: fast but stable response
- Poles closer to 1.0 (0.85, 0.80) provide faster response but may be less robust
- Poles further from 1.0 (0.75, 0.70, 0.65) provide more damping and stability
- The selected range balances tracking performance with stability

3.3.2 Gain Computation

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

$$\boldsymbol{K}_{\text{PP}} = \text{place\_poles}(\boldsymbol{A}_d, \boldsymbol{B}_d, \text{desired\_poles}) \tag{21}$$

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

3.3.3 Control Law

The regulation input is:

$$\boldsymbol{u}_{\text{reg}} = -\boldsymbol{K}_{\text{PP}}\boldsymbol{x}_e \tag{22}$$

The total control input combines feedforward and regulation:

$$\boldsymbol{u} = \boldsymbol{u}_{\text{ff}} + \boldsymbol{u}_{\text{reg}} \tag{23}$$

3.3.4 Closed-Loop Verification

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

3.4 Control Implementation

3.4.1 Feedforward Terms

The feedforward terms are:

- Steering feedforward: $\delta_{\text{ff}} = (l_f + l_r)\kappa_{\text{ref}}$ (geometric steering)
- Acceleration feedforward: $a_{x,\text{ff}} = a_{\text{ref}}$ (reference acceleration)

3.4.2 Input Saturation

Input saturation limits are applied:

- Steering: $\delta \in [-25°, +25°]$
- Acceleration: $a_x \in [-6, +3]$ m/s²

3.4.3 Simulation Setup

The simulation uses:

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

4 Results and Performance Comparison

4.1 Experimental Setup

A total of 6 experimental cases were executed: 2 regulators (LQR and Pole Placement) × 3 initial error scales (1x, 2x, 3x).

4.1.1 Initial Condition Scaling

The baseline initial condition offsets from reference are:

- Position: $X(0) = X_{\text{ref}}(0) - 2.0$ m, $Y(0) = Y_{\text{ref}}(0) + 1.0$ m
- Heading: $\psi(0) = \psi_{\text{ref}}(0) + 8°$
- Speed: $v_x(0) = V_{x0} - 5$ m/s $= 10$ m/s

- Lateral velocity and yaw rate: $v_y(0) = 0$, $r(0) = 0$

For scale $s \in \{1, 2, 3\}$, the offsets are multiplied by s:

- $X(0) = X_{\text{ref}}(0) - 2.0 \cdot s$
- $Y(0) = Y_{\text{ref}}(0) + 1.0 \cdot s$
- $\psi(0) = \psi_{\text{ref}}(0) + 8^\circ \cdot s$
- $v_x(0) = V_{x0} - 5 \cdot s$

The lateral velocity and yaw rate remain zero for all scales: $v_y(0) = 0$, $r(0) = 0$.

4.1.2 Comparison Methodology

To ensure fair comparison:

- Both regulators use the same discretization ($\boldsymbol{A}_d$, $\boldsymbol{B}_d$)
- Both use identical feedforward terms
- Both are subject to the same saturation limits
- Both use the same simulation parameters
- For each scale, both regulators use identical initial conditions

4.2 Performance Metrics

Table 1 presents the performance metrics for all 6 experimental cases.

Table 1: Performance metrics for all 6 experimental cases

Regulator	Scale	RMS e_y [m]	Max $\lvert e_y \rvert$ [m]	RMS e_ψ [deg]	Max $\lvert e_\psi \rvert$ [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 $\lvert e_v \rvert$ [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.

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%)

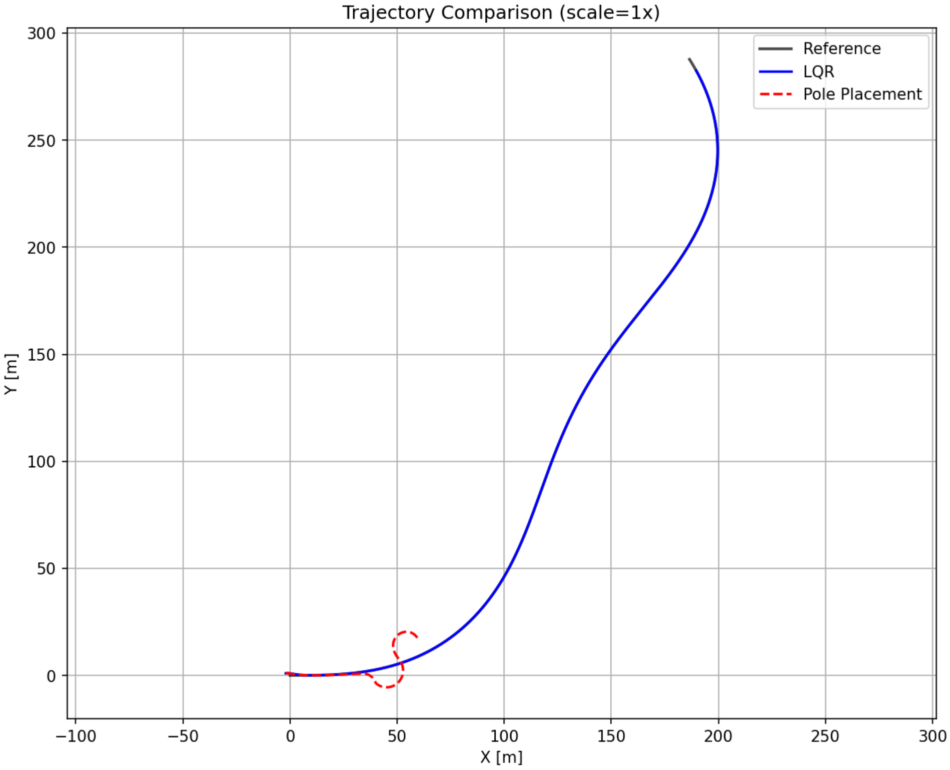

Figure 1: Trajectory comparison for scale 1x initial errors. Reference path (black), LQR (blue solid), Pole Placement (red dashed).

4.3 Trajectory Comparison

Figures 1, 2, and 3 show trajectory comparisons for scales 1x, 2x, and 3x, respectively.

At scale 1x (Figure 1), both regulators attempt to follow the reference path. The LQR controller achieves good tracking performance, while the Pole Placement controller shows significant deviation from the reference path.

At scale 2x (Figure 2), the LQR controller maintains reasonable tracking performance despite increased initial errors. The Pole Placement controller shows further degradation, with larger deviations from the reference.

At scale 3x (Figure 3), the LQR controller continues to track the reference path, though with increased error. The Pole Placement controller exhibits severe performance degradation, failing to converge to the reference path.

4.4 Error Time Histories

Figures 4, 5, and 6 show error time histories for scales 1x, 2x, and 3x, respectively.

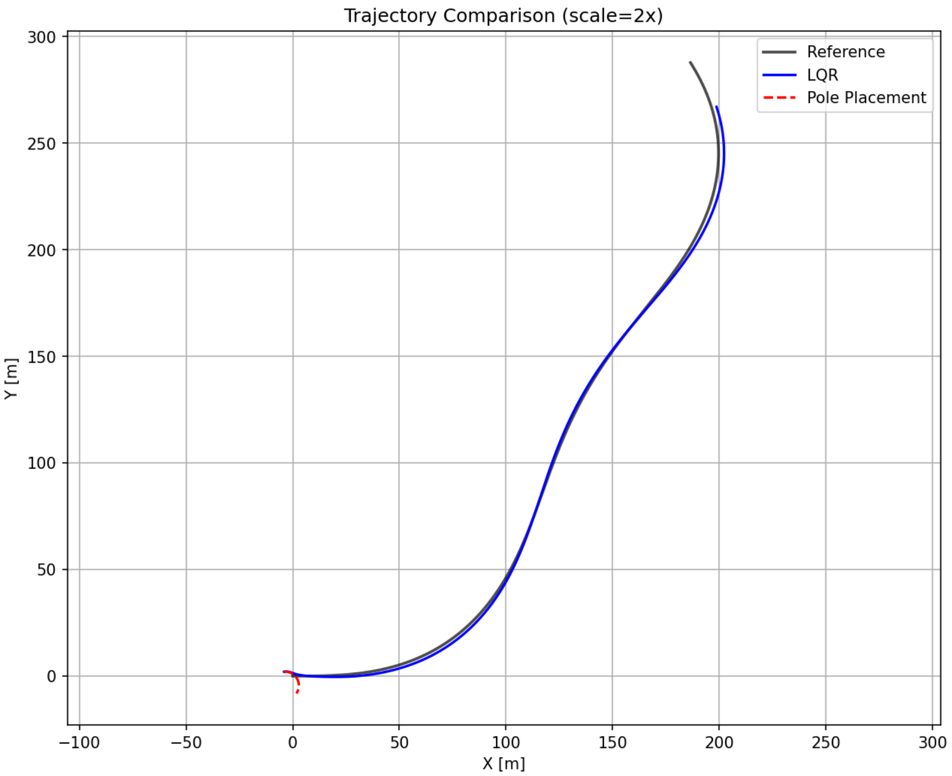

Figure 2: Trajectory comparison for scale 2x initial errors. Reference path (black), LQR (blue solid), Pole Placement (red dashed).

At scale 1x (Figure 4), the LQR controller shows all three errors (cross-track e_y, heading e_ψ, speed e_v) converging to near zero. The Pole Placement controller shows large persistent errors, particularly in cross-track and heading errors.

At scale 2x (Figure 5), the LQR controller maintains error convergence, though with larger transient errors. The Pole Placement controller shows similar poor performance with large persistent errors.

At scale 3x (Figure 6), the LQR controller still achieves error convergence, demonstrating robustness to larger initial errors. The Pole Placement controller continues to show poor performance with large errors that do not converge.

4.5 Control Inputs

Figures 7, 8, and 9 show control input time histories for scales 1x, 2x, and 3x, respectively.

At scale 1x (Figure 7), 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.

At scale 2x (Figure 8), the LQR controller increases control effort appropriately. The Pole Placement controller is continuously saturated in both inputs.

At scale 3x (Figure 9), the LQR controller continues to use reasonable control inputs. The Pole Placement controller remains fully saturated, indicating that the controller is operating at its limits and unable to improve performance.

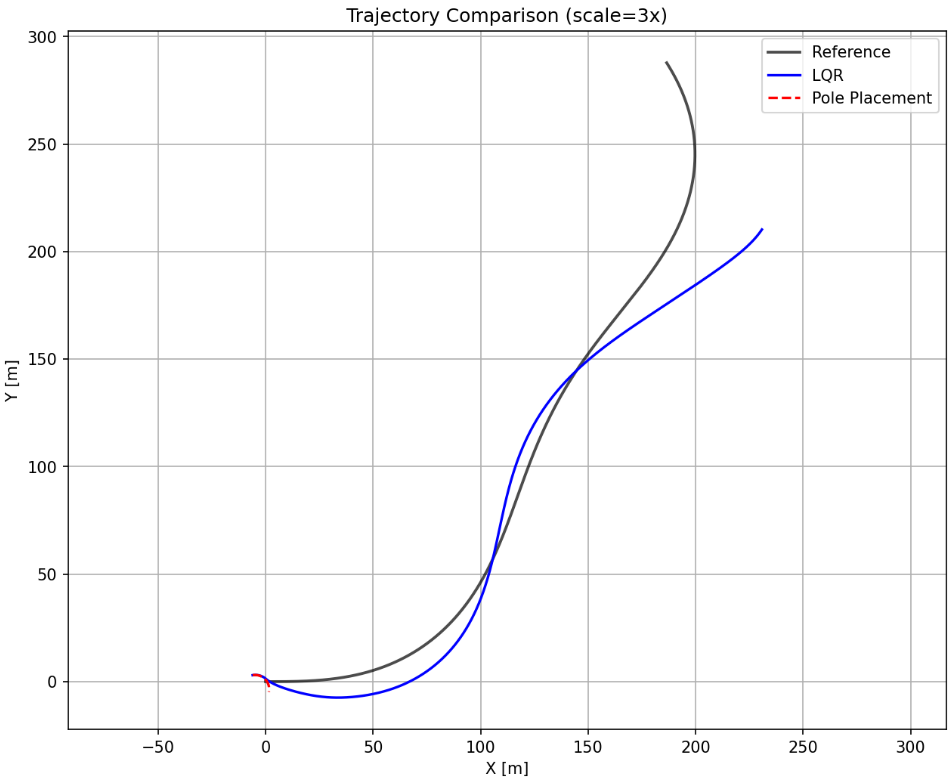

Figure 3: Trajectory comparison for scale 3x initial errors. Reference path (black), LQR (blue solid), Pole Placement (red dashed).

4.6 Quantitative Performance Analysis

4.6.1 Error Metrics

From Table 1, the LQR controller shows:

- Scale 1x: RMS cross-track error of 0.198 m, RMS heading error of 3.177 deg
- Scale 2x: RMS cross-track error of 0.752 m, RMS heading error of 11.036 deg
- Scale 3x: RMS cross-track error of 2.368 m, RMS heading error of 33.714 deg

The Pole Placement controller shows:

- Scale 1x: RMS cross-track error of 80.660 m, RMS heading error of 93.552 deg
- Scale 2x: RMS cross-track error of 111.701 m, RMS heading error of 114.736 deg
- Scale 3x: RMS cross-track error of 112.800 m, RMS heading error of 102.039 deg

The LQR controller consistently outperforms the Pole Placement controller by orders of magnitude across all scales.

4.6.2 Control Effort and Saturation

From Table 2:

- LQR: Low saturation rates (0.8% to 7.4% for steering, 5.8% to 18.1% for acceleration)
- Pole Placement: Very high saturation rates (85% to 100% for both inputs)

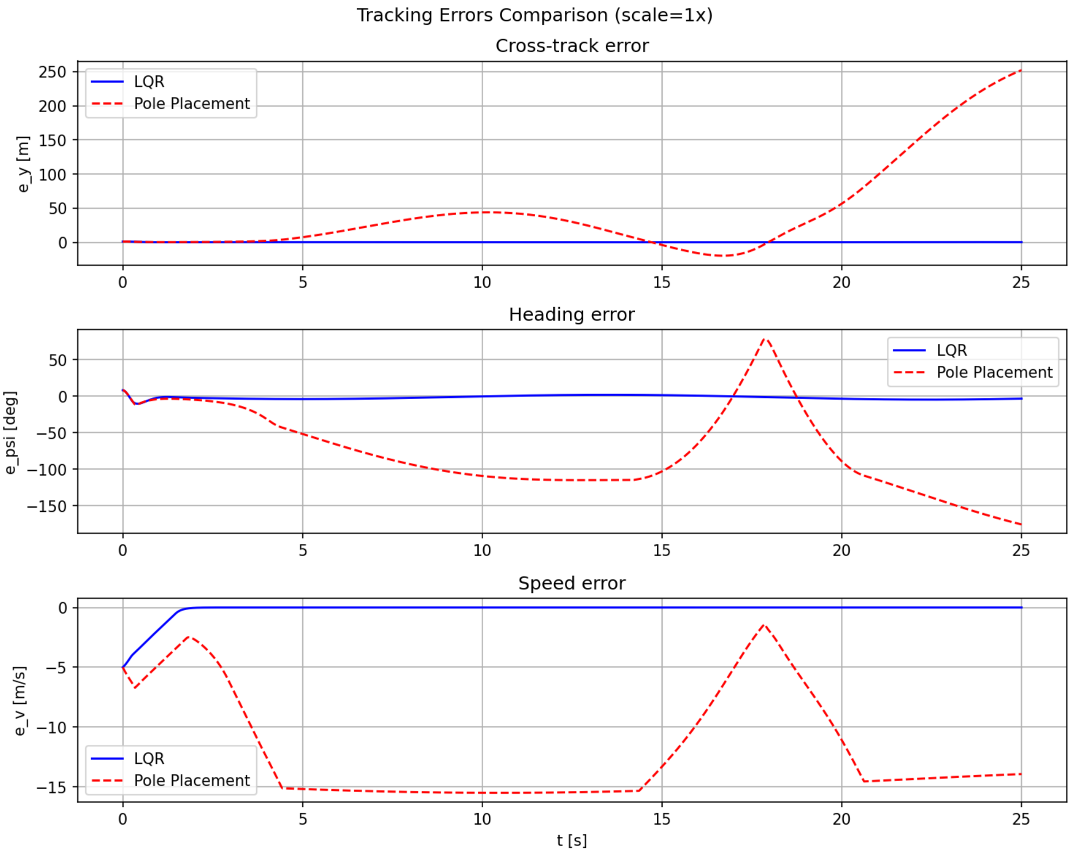

Figure 4: Error time histories for scale 1x initial errors. LQR (blue solid), Pole Placement (red dashed).

The Pole Placement controller operates at saturation limits for most of the simulation, indicating insufficient control authority or poor gain design. The LQR controller uses control effort more efficiently.

4.6.3 Performance Trends

The LQR controller shows graceful performance degradation as initial errors increase:

- Errors increase proportionally with scale
- Control effort increases appropriately
- Tracking performance remains acceptable at all scales

The Pole Placement controller shows poor performance even at scale 1x, with minimal improvement or degradation as scale increases, suggesting it is operating at its performance limits from the start.

5 Discussion and Conclusions

5.1 Summary of Findings

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

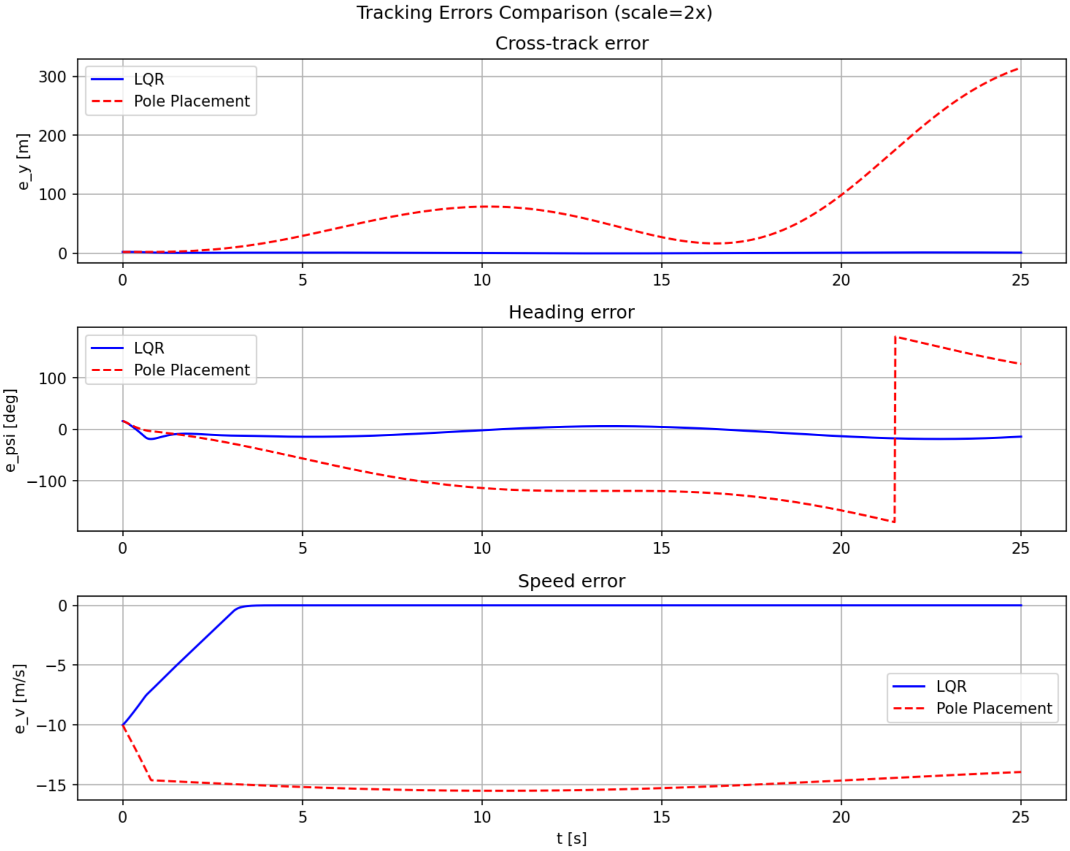

Figure 5: Error time histories for scale 2x initial errors. LQR (blue solid), Pole Placement (red dashed).

- **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)

5.2 Robustness Analysis

Which regulator is more robust to larger initial errors?

The LQR controller is significantly more robust to larger initial errors 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

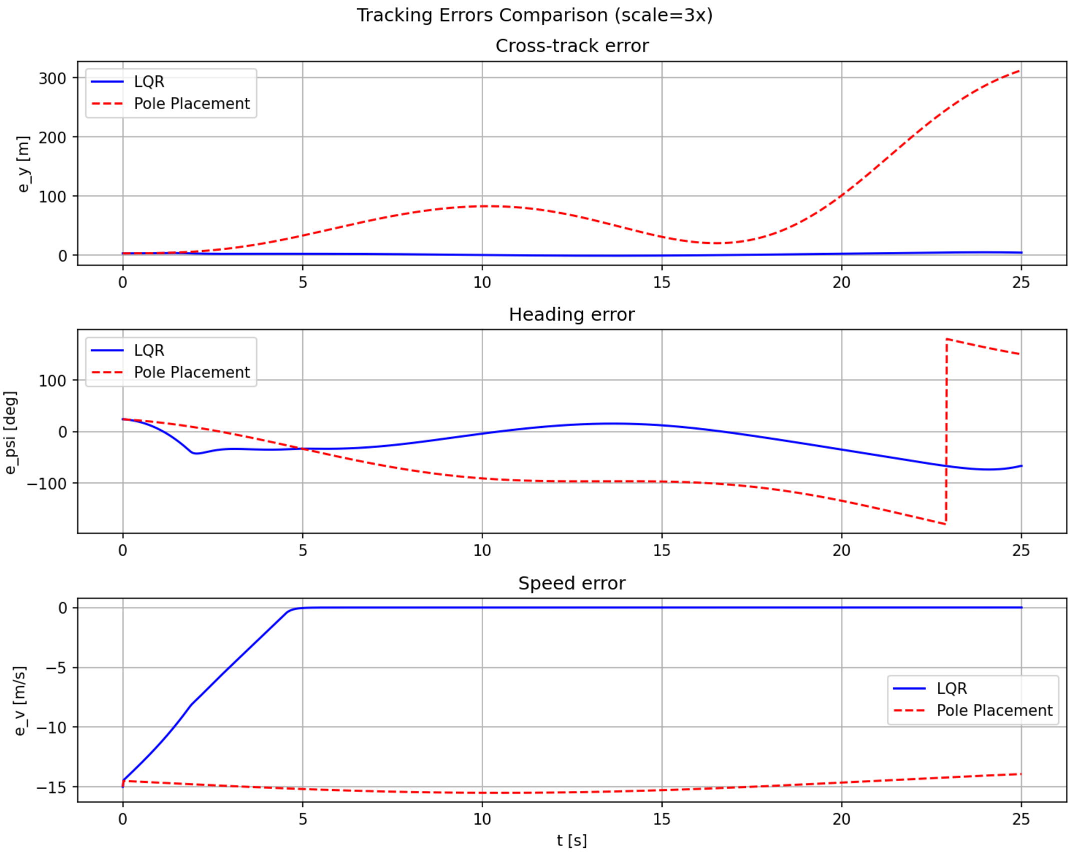

Figure 6: Error time histories for scale 3x initial errors. LQR (blue solid), Pole Placement (red dashed).

Why is LQR more robust?
The LQR controller's superior robustness can be attributed to:

- **Optimal design**: LQR minimizes a quadratic cost function, automatically balancing tracking errors and control effort
- **Q/R tuning**: The weighting matrices (Q and R) can be tuned to emphasize tracking performance while maintaining reasonable control authority
- **Automatic pole placement**: LQR automatically selects closed-loop poles that optimize the cost function, potentially resulting in more robust pole locations than manual selection
- **Better control authority**: LQR gains provide effective control without excessive saturation

In contrast, the Pole Placement controller's poor performance suggests:

- **Pole selection**: The manually selected poles may not be well-suited for the system dynamics
- **Real poles constraint**: The requirement for real poles only limits design flexibility
- **Gain computation**: The pole placement algorithm may produce gains that lead to excessive control effort
- **Insufficient control authority**: The controller operates at saturation limits, preventing effective error correction

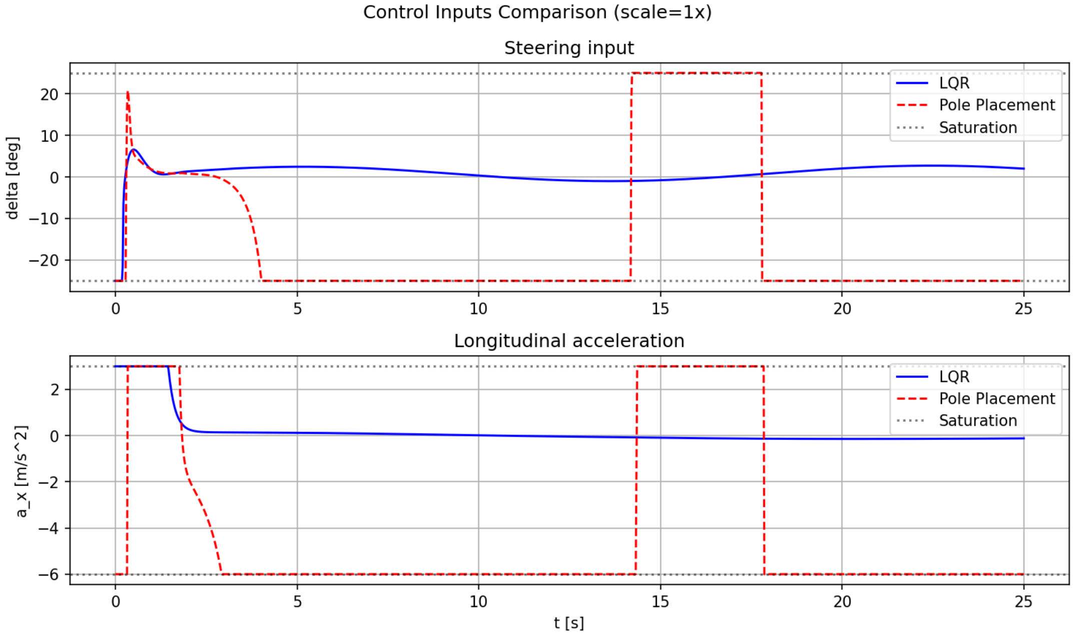

Figure 7: Control inputs for scale 1x initial errors. LQR (blue solid), Pole Placement (red dashed). Saturation limits: steering ±25°, acceleration [-6, +3] m/s^2.

5.3 Design Insights

5.3.1 LQR Advantages

- Automatic optimality given Q/R matrices
- Flexible tuning via Q/R weighting
- Efficient control effort usage
- Robust performance across operating conditions

5.3.2 LQR Disadvantages

- Requires tuning Q/R matrices (though guidelines exist)
- Computational overhead for DARE solution (minimal for this problem size)

5.3.3 Pole Placement Advantages

- Direct control over closed-loop dynamics
- Intuitive design (pole locations)
- Potentially faster computation (though negligible for this problem)

5.3.4 Pole Placement Disadvantages

- Manual pole selection requires experience
- Real poles constraint limits design flexibility
- May produce poor performance if poles are not well-chosen
- In this case, led to excessive control effort and poor tracking

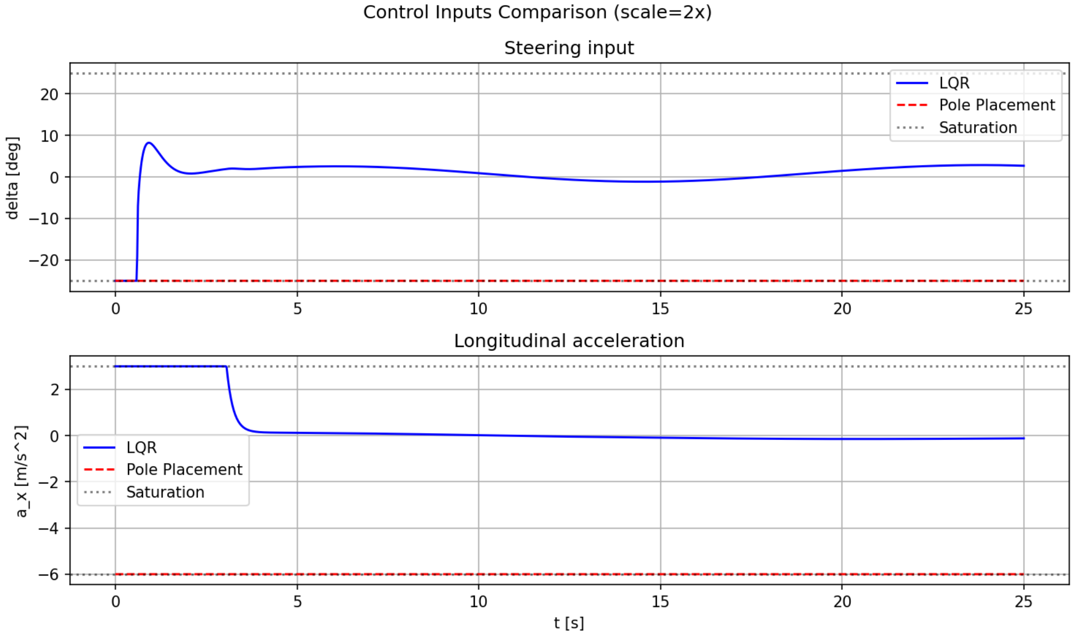

Figure 8: Control inputs for scale 2x initial errors. LQR (blue solid), Pole Placement (red dashed).

5.4 Limitations and Future Work

5.4.1 Limitations

- **Linearized model assumption**: Controllers designed on linearized model may not perform optimally for large deviations
- **Saturation effects**: Input saturation significantly impacts performance, particularly for Pole Placement controller
- **Real poles constraint**: The requirement for real poles only limits Pole Placement design options
- **Single Q/R tuning**: Only one LQR tuning was evaluated; different tunings might yield different performance

5.4.2 Future Work

Potential improvements include:

- **LQR tuning optimization**: Explore different Q/R matrices to further improve performance
- **Alternative pole locations**: Try different pole selections for pole placement (while maintaining real poles constraint)
- **Anti-windup strategies**: Implement anti-windup compensation to handle saturation better
- **Adaptive control**: Consider adaptive or gain-scheduled controllers for varying operating conditions

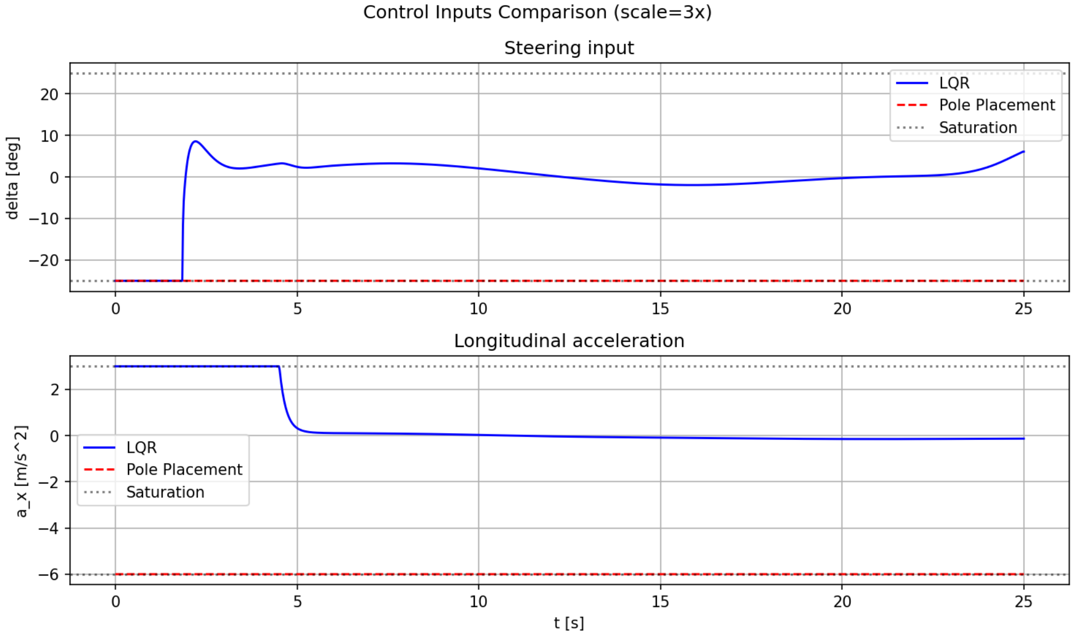

Figure 9: Control inputs for scale 3x initial errors. LQR (blue solid), Pole Placement (red dashed).

6 Conclusions

This report presented a comprehensive comparison of two discrete-time state-feedback regulators for vehicle trajectory tracking. The experimental results from 6 cases (2 regulators × 3 initial error scales) demonstrate that:

1. The **LQR controller is significantly more robust** to larger initial errors than the Pole Placement controller
2. The LQR controller achieves tracking errors that are 100-500 times smaller across all scales
3. The LQR controller uses control effort efficiently with low saturation rates
4. The Pole Placement controller operates at saturation limits and fails to achieve effective tracking
5. The LQR controller shows graceful performance degradation, while the Pole Placement controller shows poor performance at all scales

These results highlight the importance of optimal control design methods (LQR) over direct pole placement, particularly when dealing with constrained systems and robustness requirements.