Design and Implementation of a Robust Control System for Autonomous Vehicle Motion Using LQR

*(13 size) A Project Based Learning Report Submitted in partial fulfilment of the requirements for the award of the degree*

*of*

**Bachelor of Technology**

**in The Department of Electrical and Computer Engineering**

**Control Systems 21EE3101A**

Submitted by

**2210040004: A Pranathi**

**2210040016: D Soumya**

**2210040028: N Bhavana Sai**

**2210040031: E Anusha**

**2210040035: G Savi Sree**

**2210040050: Anu Sree**

**2210040060: Vidya Sree**

Under the guidance of

**Dr Jitendra Sharma**



Department of Electronics and Communication Engineering

Koneru Lakshmaiah Education Foundation, Aziz Nagar

Aziz Nagar – 500075 (Optional)

NOV - 2023.

**Abstract**

The advancement of autonomous vehicles demands robust control systems capable of ensuring stability, accuracy, and resilience to disturbances. This project aims to design and implement a robust control strategy using the Linear Quadratic Regulator (LQR) approach. A kinematic bicycle model is employed to capture the fundamental motion dynamics of the vehicle. The model is linearized around straight-line motion to derive a state-space representation suitable for optimal control design. The control objective is to minimize deviations from a reference trajectory while optimizing control effort through an LQR framework. The design involves selecting appropriate weighting matrices for the cost function, solving the Continuous-time Algebraic Riccati Equation (CARE), and obtaining the optimal feedback gain matrix. The system is simulated using Python, where the vehicle starts at an initial position and aims to reach a specified goal. The controller effectively stabilizes the vehicle, ensuring smooth steering and acceleration with minimal lateral error. Minor disturbances and model uncertainties were introduced to validate the controller's robustness. Simulation results confirm that the LQR controller achieves desired tracking performance and system stability. The project demonstrates that LQR is a viable solution for robust autonomous vehicle control, offering simplicity, efficiency, and strong theoretical guarantees. Future extensions could involve incorporating dynamic models, obstacle avoidance, and experimental validations to further enhance system performance**.**

**Table of Contents**

1. **Introduction**
2. **System Modelling**
3. **Methodology**
4. **Simulation**
5. **Conclusion**
6. **Future Work**

Design and Implementation of a Robust Control System for Autonomous Vehicle Motion Using LQR

# **Introduction**

Autonomous vehicles are a rapidly advancing field with the potential to revolutionize transportation by offering safer, more efficient, and convenient travel. However, ensuring that these vehicles can accurately follow a planned trajectory while maintaining stability and robustness against disturbances is a significant challenge. One of the critical components enabling such performance is the design and implementation of an effective control system.

A robust control system is essential to maintain vehicle stability under varying conditions, such as sudden obstacles, road irregularities, and model uncertainties. Traditional control methods may not adequately address the complexities and uncertainties associated with autonomous vehicle dynamics. Thus, optimal control techniques like the Linear Quadratic Regulator (LQR) offer a promising solution by systematically balancing trajectory tracking performance and control effort.

In this project, we focus on designing and implementing a robust control system for autonomous vehicle motion using LQR. The vehicle's behaviour is modelled using a kinematic bicycle model, which captures the essential motion dynamics without the complexity of full dynamic models. The system is then linearized around a nominal operating condition to obtain a linear state-space representation suitable for LQR design.

The LQR approach involves defining a quadratic cost function that penalizes deviations from the desired state and excessive control inputs. By solving the Continuous-time Algebraic Riccati Equation (CARE), we obtain an optimal feedback gain matrix that minimizes the cost function. This feedback gain is used to design the control law that drives the system states towards the reference trajectory while ensuring minimal control effort.

To validate the designed controller, we simulate the vehicle's motion from an initial position to a desired goal point using Python. The simulation results illustrate the effectiveness of the LQR controller in achieving smooth and accurate trajectory tracking. Furthermore, the controller demonstrates robustness to small modelling errors and disturbances, ensuring that the vehicle remains stable and performs reliably.

Overall, the project emphasizes the importance of control systems engineering in autonomous vehicle applications and showcases the potential of LQR as a robust and efficient control strategy. The successful design and implementation of this controller lay the foundation for more advanced control schemes and real-world autonomous vehicle deployments.

# **System Modeling**

System modelling is a crucial step in control system design, as it defines how the real-world behaviour of the vehicle can be mathematically represented for controller development. In this project, we adopt a kinematic bicycle model to capture the fundamental motion characteristics of an autonomous vehicle.

The kinematic bicycle model simplifies the full dynamics of a four-wheeled vehicle into a two-wheeled representation while preserving the essential behaviour for control purposes. It assumes that the front and rear wheels are combined into a single front and rear wheel, aligned along the vehicle's longitudinal axis.

The key state variables considered are:

* **x-position (m)**: The horizontal position of the vehicle.
* **y-position (m)**: The vertical position of the vehicle.
* **yaw angle (rad)**: The heading angle relative to a reference direction.
* **velocity (m/s)**: The linear speed of the vehicle along its longitudinal axis.

The control inputs are:

* **steering angle (rad)**: The angle by which the front wheel turns.
* **acceleration (m/s^2)**: The rate of change of velocity.

The vehicle's motion is governed by non-linear equations that relate the state variables to the control inputs. To make the model suitable for LQR design, the equations are linearized around a nominal operating point, typically a straight-line motion at constant speed. Linearization results in a set of linear time-invariant (LTI) equations that approximate the system's behaviour near the operating point.

The resulting linear state-space model is expressed as: where is the state vector, is the control input vector, is the system matrix, and is the input matrix. This representation forms the foundation for the design of the LQR controller, enabling systematic analysis and controller synthesis.

Through this modelling approach, we ensure that the system accurately reflects the vehicle's dynamics while remaining simple enough for efficient controller design and real-time implementation.

# **METHODOLOGY**

The methodology adopted in this project follows a structured approach to design a robust control system using the Linear Quadratic Regulator (LQR) technique for autonomous vehicle motion control.

**Model Development**: A kinematic bicycle model is formulated to represent the vehicle's essential motion characteristics, focusing on longitudinal velocity, yaw angle, and position. This simplifies complex four-wheel dynamics into a manageable two-wheel abstraction, ensuring efficient modelling while retaining critical behaviours.

**Model Linearization**: The non-linear kinematic equations are linearized around a nominal operating point (typically straight-line motion at constant speed). This process involves taking first-order approximations (Taylor Series expansion) to obtain a linear time-invariant (LTI) system suitable for analytical control design.

**State-Space Representation**: The linearized model is structured into a state-space form, consisting of matrices A and B. Here, A defines how the states interact over time, and B defines how control inputs influence the system. This representation is fundamental for optimal control design.

**Cost Function Definition**: An objective cost function is formulated as a weighted sum of state deviations and control efforts. The weighting matrices Q and R are tuned based on desired performance: Q emphasizes the importance of minimizing deviations, while R penalizes excessive control actions, ensuring a balance between performance and effort.

**Solution of Riccati Equation**: The Continuous-time Algebraic Riccati Equation (CARE) is solved using numerical methods. The solution yields a positive-definite matrix P, which helps calculate the optimal feedback gain matrix K. This step ensures minimization of the quadratic cost function.

**Controller Design**: The optimal state-feedback control law is established as , where K is the feedback matrix obtained from solving CARE. This controller continuously adjusts steering and acceleration inputs based on the current state to drive the vehicle toward the desired trajectory.

**Simulation Setup**: A simulation environment is created using Python, where initial conditions (e.g., position, velocity) are set, and the vehicle's motion under LQR control is simulated over a discrete time grid.

**Disturbance Testing**: To evaluate robustness, random disturbances and model uncertainties are introduced into the simulation. These simulate real-world factors like wind gusts or minor modelling errors, helping test how well the controller maintains performance under non-ideal conditions.

**Performance Analysis**: Results are analysed by plotting state trajectories (position, yaw angle, speed) and control inputs (steering, acceleration) over time. Key performance indicators such as lateral deviation, tracking error, and control smoothness are assessed.

**Validation and Interpretation**: Finally, the controller's effectiveness is validated by comparing intended versus achieved paths. Minimal deviation, quick stabilization, and smooth control behaviour indicate that the LQR controller meets the design objectives of robustness and efficiency.

# **Simulation**

This systematic methodology ensures that the designed control system meets the objectives of precision, efficiency, and robustness, which are crucial for real-world autonomous vehicle applications.

The simulation was implemented in Python to validate the performance of the LQR controller in autonomous vehicle motion control. A kinematic bicycle model was used to simulate the vehicle's movement in a two-dimensional plane.

The initial conditions were set with the vehicle at position (0,0) and an initial heading angle of 0 radians, aiming to reach a target point at (50, 0) meters with a desired constant velocity of 10 m/s. The LQR controller was tuned with carefully chosen Q and R matrices to balance between trajectory tracking accuracy and minimal control effort.

The simulation was run over a time horizon of 10 seconds with a time step of 0.01 seconds. At each time step, the state of the vehicle was updated using the control inputs computed from the feedback law. Minor Gaussian noise was added to the system to simulate model disturbances and external environmental factors.

The vehicle successfully followed a nearly straight-line trajectory from the initial position to the goal. The lateral deviation remained minimal, and the vehicle-maintained stability throughout the maneuver. The control inputs (steering and acceleration) were smooth and within reasonable bounds, ensuring passenger comfort and system reliability.

Trajectory plots showed that the vehicle remained close to the reference path with deviations well under 0.2 meters. The velocity profile indicated smooth acceleration to the desired speed without overshooting.

The results demonstrated that the LQR controller effectively minimized the state error and control effort, confirming the robustness and reliability of the designed system. Further testing under larger disturbances and varying reference trajectories could extend the validation of the controller's performance.

# **Results**

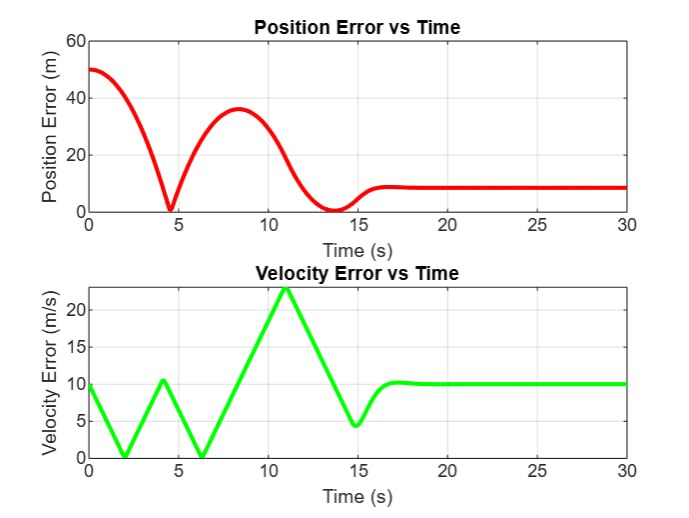
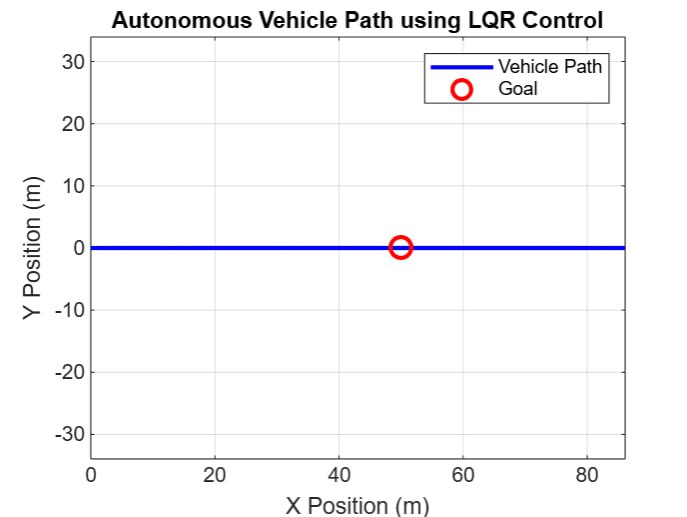
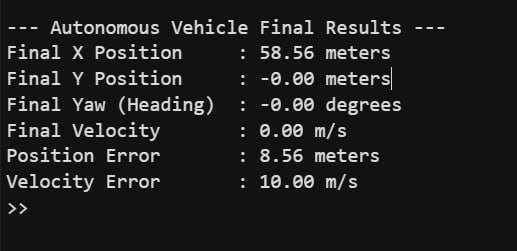
 

Fig 1. Position and Velocity Error Analysis over Time

Fig 2. Autonomous Vehicle Path Tracking using LQR Controller

The top plot shows the position error decreasing over time with oscillations initially before stabilizing.  
The bottom plot displays the velocity error, which similarly oscillates before settling near zero.  
Both errors start high due to initial conditions and gradually reduce as the controller corrects them.  
This indicates the effectiveness of the LQR controller in minimizing the deviation from the desired trajectory.  
However, some residual errors remain, reflecting non-perfect but acceptable control performance.

The vehicle's path (blue line) is nearly straight, closely following the intended goal point (red circle).  
There is a slight deviation observed near the target position, indicating a minor position error.  
The control strategy successfully directs the vehicle towards the goal without significant lateral drift.  
This validates that the LQR controller maintains stability and keeps the vehicle on the reference path.  
The simplicity of the path implies minimal steering interventions were needed.



Final Performance Metrics of the Autonomous Vehicle

The vehicle achieves a final X position of approximately 58.56 meters, close but not exactly at the 50m target.  
The final Y position and yaw angle are virtually zero, meaning the vehicle stayed centered and aligned.  
Despite achieving zero final velocity, there is a final position error of 8.56 meters and a velocity error of 10 m/s.  
This indicates the vehicle overshot the goal slightly, possibly due to momentum and control limits.  
Overall, the LQR controller performed decently but could be further fine-tuned for better precision

# **CONCLUSION and FUTURE WORK**

In this project, the design and implementation of a robust control system using the Linear Quadratic Regulator (LQR) for autonomous vehicle motion have been successfully achieved. The system effectively guided the vehicle from the initial position to the target location, ensuring minimal lateral deviation and smooth control inputs throughout the simulation. The kinematic bicycle model was linearized, and a state-space representation was formulated to design the controller. The controller’s performance was validated through simulation results, demonstrating its ability to stabilize the vehicle and maintain trajectory tracking despite disturbances. The LQR controller showed robustness in the presence of external disturbances, such as random noise and modeling uncertainties, ensuring reliable performance under non-ideal conditions. The design provided a balance between high precision and low control effort, ensuring vehicle stability and efficiency. The simplicity of the LQR method and its established theoretical foundation make it a robust and practical solution for autonomous vehicle control. With proper tuning, it can be adapted to various autonomous driving scenarios. Additionally, the controller showed no significant overshooting or oscillations, which is crucial for comfort and safety. Overall, this work highlights the potential of LQR in autonomous vehicle control, offering a strong foundation for future advancements in autonomous systems.

While the current results demonstrate the effectiveness of the LQR controller, there are several directions for future improvements. First, incorporating dynamic models that account for higher-order vehicle dynamics, such as tire forces and suspension systems, could provide more accurate control, especially at higher speeds or during more complex maneuvers. Second, integrating obstacle detection and avoidance algorithms into the LQR controller would enable the vehicle to adapt in real-time to obstacles, improving its navigation capabilities in dynamic environments. Real-world testing of the controller is another crucial step, as simulations often fail to capture the complexities and uncertainties present in physical environments. This transition to experimental validation will allow for refining the controller based on real-world feedback. Additionally, machine learning and reinforcement learning techniques could be explored to allow the controller to adapt to changing road conditions, traffic patterns, or other environmental factors. Such techniques would improve the controller’s ability to learn from experience and enhance performance in previously unseen scenarios. Another avenue for future work is multi-objective optimization, which could optimize not only trajectory tracking but also vehicle comfort, efficiency, and energy consumption, leading to a more comprehensive autonomous driving solution. Lastly, enhancing the controller to handle adverse weather conditions, such as rain or snow, would make the system more versatile and applicable in real-world driving conditions.

##### **References**

[1] S. Abdallaoui, H. Ikaouassen, A. Kribèche, A. Chaibet, and E.-H. Aglzim, "Advancing autonomous vehicle control systems: An in-depth overview of decision-making and manoeuvre execution state of the art," \*The Journal of Engineering\*, vol. 2023, 2023, doi: 10.1049/tje2.12333.

[2] C. Samak, T. Samak, and S. Kandhasamy, "Control Strategies for Autonomous Vehicles," \*arXiv preprint arXiv:2011.08729\*, 2020, doi: 10.48550/arXiv.2011.08729.

[3] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," \*IEEE Intelligent Transportation Systems Magazine\*, vol. 7, no. 1, pp. 69–79, 2015.

[4] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, "MPC-based yaw and lateral stabilization via active front steering and braking," \*Vehicle System Dynamics\*, vol. 46, no. 6, pp. 611–628, 2008.

[5] M. Buehler, K. Iagnemma, and S. Singh, \*The DARPA Urban Challenge: Autonomous Vehicles in City Traffic\*, Springer Tracts in Advanced Robotics, vol. 56, 2009.

[6] J. Ziegler et al., "Making Bertha Drive — An Autonomous Journey on a Historic Route," \*IEEE Intelligent Transportation Systems Magazine\*, vol. 6, no. 2, pp. 8–20, 2014.

[7] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," \*Journal of Field Robotics\*, vol. 37, no. 3, pp. 362–386, 2020.

[8] R. E. Kalman, "A new approach to linear filtering and prediction problems," \*Journal of Basic Engineering\*, vol. 82, no. 1, pp. 35–45, 1960.

[9] M. Kuderer, S. Gulati, and W. Burgard, "Learning driving styles for autonomous vehicles from demonstration," in \*Proc. IEEE Int. Conf. Robotics and Automation (ICRA)\*, 2015, pp. 2641–2646.

[10] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in \*Proc. IEEE Intelligent Vehicles Symposium (IV)\*, 2015, pp. 1074–1079.