Lane Keeping Assist System(LKAS)

Applied Automatic Control
Advanced Automotive Engineering
Electronic Engineering for Intelligent Vehicles

University of Bologna

A.A. 2025–2026

Kartik Purushottam Kanchan and Edoardo Guglielmi

December 22, 2025

Contents

Abstract

This project for the Applied Automatic Control course is based on the IEEE paper "A Comprehensive Predictive Control for Lane Keeping Assistance Systems." The vehicle dynamics are represented using a two-degree-of-freedom bicycle model provided in the reference study. The lane-keeping control strategy relies exclusively on information obtained from a forward-looking camera, which provides lane-related measurements such as lateral displacement, heading error, and road curvature. Using these camera-based signals as inputs, a feedback control approach derived from classical control techniques studied in the course is implemented to minimize tracking errors and ensure accurate path following in the presence of disturbances. By compensating for vehicle lateral dynamics using vision-based feedback only, the proposed strategy reduces driver workload while improving ride comfort and safety. Simulation results demonstrate the effectiveness of the implemented controller in maintaining lane tracking performance and vehicle stability.

1 Introduction

Road safety relies on a combination of regulations, driver awareness, and onboard technologies aimed at reducing accidents before they occur. In recent years, vehicle manufacturers and public authorities have increasingly focused on systems that support the driver during critical driving tasks. Active safety functions such as ABS, ESC, emergency braking, and lane assistance act directly on the vehicle to limit unsafe situations, often compensating for distraction, fatigue, or delayed reactions. Within this context, lane assistance plays a key role in lateral vehicle control, especially on long trips and high-workload road segments.

The Lane Keeping Assist System supports the driver by monitoring lane boundaries through onboard sensors and applying small steering corrections to keep the vehicle centered. Its effectiveness, however, is strongly influenced by sensor reliability, environmental conditions, and the quality of the control strategy. Poor visibility, degraded road markings, or overly aggressive steering actions can reduce comfort and lead to system deactivation. For this reason, control design remains a central challenge. This report investigates Model Predictive Control for LKAS, focusing on its ability to handle constraints, anticipate vehicle behavior, and achieve smooth, reliable lane keeping under realistic operating conditions.

Lane Keeping Assist System (LKAS) solutions detect unintentional lane drift by adjusting the steering to keep the vehicle within its lane. A forward-looking camera serves as a sensor input for LKAS, detecting lines and lane features. Information from the car driving scene, such as lateral offset, heading angle, lane curvature, etc., is retrieved and forwarded to the LKAS controller. This information is used to make subtle adjustments to maintain the vehicle in the middle of the detected lane. Additional inputs, such as sidelight activation and hands on the steering wheel, are checked to avoid unauthorized vehicle intervention.

LKAS provides a more confident driving experience on narrow roadways. It often reduces unintentional lane departure resulting from driver fatigue, high workload, momentary distraction, or somnolence.Through an adaptive learning mechanism, LKAS adjusts itself, taking into account the driver's style and avoiding potential conflicts between steering actuation and driver intervention. LKAS proves particularly useful in long trips or heavy traffic conditions.

2 Vehicle Model and System Description

The lateral vehicle dynamics will be fully described using a two-degree-of-freedom (2-DOF) bicycle model (see Figure 1 taken from A Comprehensive Analysis of Model Predictive Control for Lane Keeping Assist System).The model in this paper is simplified, by making certain assumptions that have been made, that includes, the exclusion of roll and pitch dynamics, and the use of small angles and constant longitudinal speed. In such scenarios, the assumption of linearity in the bicycle model may no longer be valid. However, concerning LKAS, simplifying model conditions doesn't considerably interfere with the vehicle model behaviour. This is primarily because, in most cases, the vehicle dynamics operate within a linear operating zone, that reduce impact of model simplification.

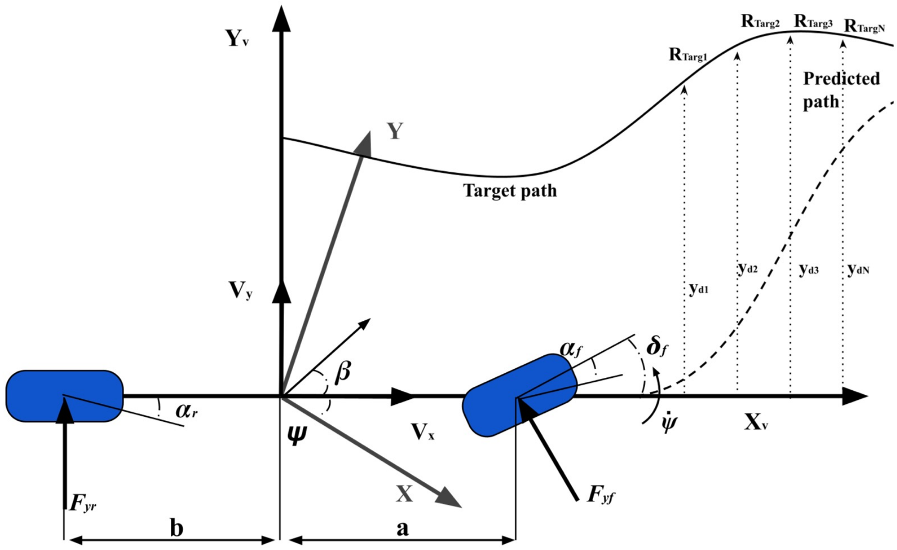

Figure 1: Bicycle vehicle model

The main parameters table below describe in details all the variables used:

Table of main parameters

Table 1: Vehicle Parameters

Symbol	Description	Value
m	Vehicle body mass	1573 kg
I_z	Moment of inertia around z-axis	2873 kg m^2
a	front axles distance from CG	1.1 m
b	rear axles distance from CG	1.58 m
C_{α_f}	Front cornering stiffness	80.000 N/m
C_{α_r}	Rear cornering stiffness	80.000 N/m
V_x	Logitudinal vehicle speed	16.667 m/s

2.1 Equation for state

The lateral vehicle position and yaw angle are represented by 2-DOF model and are derived from the following differential equation:

$$\begin{aligned} ma_y &= -mv_x\dot{\psi} + F_{Fy} + F_{Ry} \\ I_z\ddot{\psi} &= l_f F_{Fy} - l_r F_{Ry} \end{aligned} \tag{1}$$

The vehicle heading or yaw angle evolves according to the yaw angular velocity about the vertical z-axis. Therefore, the yaw rate is introduced as:

$$\dot{\psi} = r_\omega \tag{2}$$

substitute equation (1) with: $a_y = \dot{v_y}$ and $\dot{\psi} = r_\omega$

$$\begin{aligned} m\dot{v_y} &= -mv_x\dot{\psi} + F_{Fy} + F_{Ry} \\ I_z\dot{r_\omega} &= aF_{Fy} - bF_{Ry} \end{aligned} \tag{3}$$

Where m represents the vehicle mass;ψ,$\dot{\psi}$ and $\ddot{\psi}$ devote the yaw angle, speed and acceleration about the z-axis at the center of gravity (CG) of the vehicle; a_y represents the lateral acceleration; v_y represents the lateral velocity; v_x is the longitudinal vehicle speed. The lateral tire force F_{Fy} and F_{Ry} can be expressed by following equation:

$$\begin{aligned} F_{Fy} &= 2C_{\alpha_f}\alpha_f \\ F_{Ry} &= 2C_{\alpha_r}\alpha_r \end{aligned} \tag{4}$$

In the following expressions, the slip angle (α) is crucial for computing the wheel direction and the true vehicle motion. The front (α_f) and rear (α_r) slip angles are defined as follows, where the front axle is actuated by the steering input δ:

$$\alpha_f = \delta - \frac{v_y + a\dot{\psi}}{v_x}; \qquad \alpha_r = -\frac{v_y - b\dot{\psi}}{v_x}. \tag{5}$$

The following equation describes the lateral kinematic relationship of the vehicle, where y_d denotes the lateral displacement and its time derivative represents the lateral velocity governing the motion required to reduce the deviation between the actual vehicle path and the reference lane.

$$\dot{y_d} = v_y + v_x\psi \tag{6}$$

2.2 Model and Problem Formulation

Using the above-mentioned equations (2), (3), and (6), the system is represented in state-space form with the state vector defined as:

$$x = \begin{bmatrix} v_y \\ r_\omega \\ \psi \\ y_d \end{bmatrix}; \qquad u = \delta; \qquad d = \begin{bmatrix} C_{\alpha_f} \\ C_{\alpha_r} \end{bmatrix}; \qquad w = \begin{bmatrix} d \\ \nu \\ r \end{bmatrix} \tag{7}$$

Here:

- x is the state vector, that consist of lateral velocity v_y, yaw rate r_ω, heading angle ψ, and lateral position y_d.

- u is the control input vector containing the steering angle δ at front axle.
- ν represents sensor noise of camera used to find lateral displacement y_d.
- d is the disturbance vector, representing external inputs such as cornering stiffness.
- w is the complete vector of exogenous signals, including disturbances, noise, and the reference in this case path.

3 State Space and Linearization

Considering oriented model we can define:
$\dot{x} = f(x, u, w)$; $y = h(x, u, w)$; $e = h_e(x, u, w)$
where:

- $x : \mathbb{R} \to \mathbb{R}^n$, with $n \in \mathbb{N}$, called the state vector.
- $u : \mathbb{R} \to \mathbb{R}^p$, with $p \in \mathbb{N}$, called the control input vector.
- $w : \mathbb{R} \to \mathbb{R}^r$, with $r \in \mathbb{N}$, defined as $w := \mathrm{col}(d, \nu, r)$, called the vector of exogenous signals.
- $y : \mathbb{R} \to \mathbb{R}^q$, with $q \in \mathbb{N}$, called the measurement output vector.
- $e : \mathbb{R} \to \mathbb{R}^m$, with $m \in \mathbb{N}$, called the regulated output or error vector.
- $f : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}^n$, called the process state model.
- $h : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}^q$, called the output function.
- $h_e : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^r \to \mathbb{R}^m$, called the error function.

Referring to the described model we will define our LKAS as follows:

$$f(x, u, w) = \begin{bmatrix} -\dfrac{2C_r\left(v_y - br_\omega\right)}{mv_x} + \dfrac{2C_f\left(u - \dfrac{v_y + ar_\omega}{v_x}\right)}{m} - v_x r_\omega \\ \dfrac{2aC_f\left(u - \dfrac{v_y + ar_\omega}{v_x}\right)}{I_z} + \dfrac{2bC_r\left(v_y - br_\omega\right)}{I_z v_x} \\ r_\omega \\ v_y + v_x\psi \end{bmatrix}$$

$$h(x, u, w) = \begin{bmatrix} y_d + \nu \end{bmatrix}$$

$$h_e(x, u, w) = \begin{bmatrix} y_d + \nu - r \end{bmatrix}$$

3.1 Linearization

Now, after finding process model, output function, and error function we will have to find $(x^\star, u^\star, w^\star)$ that is an equilibrium triplet if $f(x^\star, u^\star, w^\star) = 0$. Also, we need to find $y^\star := h(x^\star, u^\star, w^\star)$ and $e^\star := h_e(x^\star, u^\star, w^\star)$ the equilibrium output and equilibrium error. The equilibrium error $e^\star = 0$, usually considered from this we can determine:

$$\begin{aligned} e^\star := {y_d}^\star + \nu^\star - r^\star = 0 \\ \therefore {y_d}^\star = r^\star \end{aligned} \tag{8}$$

$$\dot{x}^\star = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.1 \end{bmatrix}; \qquad u^\star = \begin{bmatrix} 0 \end{bmatrix}; \qquad w^\star = \begin{bmatrix} 80,000 \\ 80,000 \\ 0 \\ 0 \end{bmatrix}; \tag{9}$$

The result of equilibrium is achieved by using the known values in Table 1.
Considering the equilibrium triplet we can define non- linear values of the states described above and then we can get the following equation.

$$\tilde{x}_{\mathrm{NL}} := x - x^\star; \qquad \tilde{u} \ := u - u^\star; \qquad \tilde{w} := w - w^\star; \qquad \tilde{y}_{\mathrm{NL}} \ := y - y^\star; \qquad \tilde{e}_{\mathrm{NL}} := e - e^\star \tag{10}$$

Finally, we can define design model as

$$\begin{aligned} \dot{\tilde{x}} &= A\tilde{x} + B_1\tilde{u} + B_2\tilde{w}, \qquad \tilde{x}(t_0) = \tilde{x}_0, \\ \tilde{y} &= C\tilde{x} + D_1\tilde{u} + D_2\tilde{w}, \\ \tilde{e} &= C_e\tilde{x} + D_{e1}\tilde{u} + D_{e2}\tilde{w}. \end{aligned} \tag{11}$$

where A, B_1, B_2, C, D_1, D_2, C_e, D_{e1}, and D_{e2} are the Jacobian matrices obtained from linearization of the nonlinear process, output, and error functions at the equilibrium point. The equation below defines the matrix according to our LKAS:

$$\begin{gathered} A = \begin{bmatrix} -\dfrac{2(C_f + C_r)}{mV_x} & -\dfrac{V_x m + \frac{2C_f a}{V_x} - \frac{2C_r b}{V_x}}{m} & 0 & 0 \\ -\dfrac{\frac{2C_f a}{V_x} - \frac{2C_r b}{V_x}}{I_z} & -\dfrac{\frac{2C_f a^2}{V_x} + \frac{2C_r b^2}{V_x}}{I_z} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & V_x & 0 \end{bmatrix}; \\ B_1 = \begin{bmatrix} \dfrac{2C_f}{m} \\ \dfrac{2aC_f}{I_z} \\ 0 \\ 0 \end{bmatrix}; \qquad B_2 = \begin{bmatrix} \dfrac{2u - \frac{2(v_y + ar_\omega)}{V_x}}{m} & -\dfrac{2(v_y - br_\omega)}{mV_x} & 0 & 0 \\ \dfrac{2a\left(u - \frac{v_y + ar_\omega}{V_x}\right)}{I_z} & \dfrac{2b(v_y - br_\omega)}{I_z V_x} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \\ C = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}; \qquad D_1 = 0; \qquad D_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}; \\ C_e = \begin{bmatrix} 0 & 0 & 0 & -1 \end{bmatrix}; \qquad D_{e1} = 0; \qquad D_{e2} = \begin{bmatrix} 0 & 0 & -1 & 1 \end{bmatrix}. \end{gathered} \tag{12}$$

Finally after using the equilibrium values we get.

$$
\begin{aligned}
A &= \begin{bmatrix} -12.2057 & -13.7376 & 0 & 0 \\ 1.6039 & -12.3845 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 1.0000 & 0 & 16.6670 & 0 \end{bmatrix}; \\
B_1 &= \begin{bmatrix} 101.7165 \\ 61.2600 \\ 0 \\ 0 \end{bmatrix}; \qquad B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}; \\
C &= \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}; \qquad D_1 = 0; \qquad D_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}; \\
C_e &= \begin{bmatrix} 0 & 0 & 0 & -1 \end{bmatrix}; \qquad D_{e1} = 0; \qquad D_{e2} = \begin{bmatrix} 0 & 0 & -1 & 1 \end{bmatrix}.
\end{aligned} \tag{13}
$$

The numerical values of the linearized system matrices are computed in MATLAB by substituting the nominal equilibrium parameters reported in Table 1, and the results are summarized in Equation (13).

4 Control Strategy: Controllability and Observability Analysis

The control feasibility of the Lane Keeping Assist System (LKAS) was assessed by analyzing the controllability and observability properties of the linearized vehicle model. These properties ensure that the steering input can effectively influence the lateral vehicle motion and that the system states can be reconstructed from the available camera-based measurements.

The controllability matrix Rm, n of the linearized system was computed in MATLAB using the `ctrb` command. For improved numerical interpretation, the matrix was normalized column by column. The rank condition confirmed that the LKAS lateral dynamics are fully controllable, indicating that the steering input is sufficient to regulate the lateral velocity, yaw rate, heading angle, and lateral displacement of the vehicle.

To account for integral action in the controller design, an augmented system was introduced. The augmented state-space matrices are defined as

$$
A_e = \begin{bmatrix} A & 0 \\ C_e & 0 \end{bmatrix}, \qquad B_{1e} = \begin{bmatrix} B_1 \\ D_{e1} \end{bmatrix}. \tag{14}
$$

The corresponding extended controllability matrix is given by

$$
R_e = \mathrm{ctrb}(A_e, B_{1e}) = \begin{bmatrix} B_{1e} & A_e B_{1e} & A_e^2 B_{1e} & \cdots & A_e^{n+q-1} B_{1e} \end{bmatrix}, \tag{15}
$$

where n and q denote the dimensions of the state and integral-error subsystems, respectively. The full rank of R_e confirms that the augmented LKAS model is fully reachable, allowing the use of state-feedback control with integral action.

Observability was evaluated by constructing the observability matrix

$$O = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}, \tag{16}$$

which was likewise normalized column by column. The rank test verified that the observability matrix is full rank, demonstrating that the system states can be uniquely reconstructed from the camera-based lateral displacement measurement.
Since the LKAS model is both controllable and observable, it is suitable for the design of a state-feedback regulator with integral action using Linear Quadratic Regulator (LQR) techniques, as well as for the synthesis of a state observer based on a Kalman filter.
In the subsequent sections, the selection of the LQR weighting matrices for lateral control and integral action is discussed, followed by the design of the Kalman observer. Finally, the closed-loop performance of the LKAS is evaluated through Simulink simulations. To assess robustness under varying driving conditions, a path reference is applied at $t = 100\,\text{s}$, emulating lane curvature variations and testing the system's tracking and stability performance.
The system is both reachable and observable and reach able for our LKAS as mentioned below:

$$\mathbf{R} = 10^5 \begin{bmatrix} 0.0010 & -0.0208 & 0.3361 & -4.6562 \\ 0.0006 & -0.0060 & 0.0403 & 0.0394 \\ 0 & 0.0006 & -0.0060 & 0.0403 \\ 0 & 0.0010 & -0.0106 & 0.2368 \end{bmatrix} \tag{17}$$

```
>> R = ctrb(Amatrix, B1matrix);

if rank(R) == length(Amatrix)
    disp('(Amatrix,B1matrix) is fully reachable because rank(R)=n')
else
    disp('(Amatrix,B1matrix) is NOT fully reachable because rank(R)<n')
end
(Amatrix,B1matrix) is fully reachable because rank(R)=n
```

Figure 2: Reachability matrix rank

Since $\text{rank}(\mathbf{R}) = 4$, which equals the system order, the pair (A, B_1) is fully controllable (reachable).

The observability of the LKAS is mentioned below:

$$O = \begin{bmatrix} 0 & 0 & 0 & 1.0000 \\ 1.0000 & 0 & 16.6670 & 0 \\ -12.2057 & 2.9294 & 0 & 0 \\ 153.6782 & 131.3988 & 0 & 0 \end{bmatrix} \tag{18}$$

Since $\text{rank}(O) = 4$ equals the system order, the pair (A, C) is fully observable. Therefore, the LKAS states can be reconstructed from the camera-based lateral displacement measurement, enabling the design of a Kalman observer.

```
>> O = obsv(Amatrix, Cmatrix);

n = size(Amatrix,1);

if rank(O) == n
    disp('(A,C) is fully observable because rank(O) = n')
else
    disp('(A,C) is NOT fully observable because rank(O) < n')
end
(A,C) is fully observable because rank(O) = n
```

Figure 3: Observability matrix rank

Observability was assessed using the observability matrix constructed from the linearized LKAS model. Despite measuring only the lateral displacement through the camera, the rank condition confirms full observability. This guarantees that all system states, including yaw rate and heading angle, can be uniquely reconstructed using a Kalman observer.

4.1 LQR State-Feedback with Integral Action

The LQR controller with integral action is implemented in MATLAB following the theoretical formulation described above. First, the augmented system matrices are constructed by extending the original state-space model with the integral of the tracking error:

$$A_e = \begin{bmatrix} A & 0 \\ C_e & 0 \end{bmatrix}, \qquad B_{1e} = \begin{bmatrix} B_1 \\ D_{e1} \end{bmatrix}.$$

To improve numerical conditioning and ensure a desired stability margin, a positive shift parameter α is introduced, yielding the modified system matrix

$$A_m = A_e + \alpha I.$$

The control input constraint is taken into account by defining the weighting matrix

$$R = \frac{1}{u_{\max}^2},$$

where $u_{\max}$ denotes the maximum allowable steering angle. The state weighting matrix Q is selected as a diagonal matrix, where each diagonal element penalizes the corresponding augmented state. In particular, a higher weight is assigned to the integral state to enforce accurate steady-state tracking:

$$Q = \left(n \cdot \operatorname{diag}\left(\epsilon_1^2, \epsilon_2^2, \epsilon_3^2, \epsilon_4^2, \epsilon_5^2\right)\right)^{-1}.$$

The optimal feedback gain is computed by solving the Continuous-time Algebraic Riccati Equation (CARE) using the `icare` algorithm:

$$A_m^\top X + X A_m - X B_{1e} R^{-1} B_{1e}^\top X + Q = 0.$$

The resulting optimal gain matrix is given by

$$K = -R^{-1} B_{1e}^{\top} X,$$

which is partitioned as

$$K = \begin{bmatrix} K_s & K_{IA} \end{bmatrix},$$

where K_s represents the state-feedback gains and K_{IA} corresponds to the integral action gain. This control structure guarantees closed-loop stability, optimal transient performance, and zero steady-state tracking error for the lateral displacement, making it suitable for Lane Keeping Assist System (LKAS) applications.

```
%% LQR state-feedback with integral action

Ae = [Amatrix  zeros(n,s);
Cematrix  zeros(s,s)];

B1e = [B1matrix;
De1matrix];

alpha = 1;            % stability margin
umax  = 2.12;         % maximum steering angle

Ceps = eye(size(Ae));

R = 1/umax^2;         % input weighting

eps1max = 1;
eps2max = 1;
eps3max = 1;
eps4max = 1;
eps5max = 0.01;       % integral state weighting

Q = inv(length(Ceps) * diag([
eps1max^2;
eps2max^2;
eps3max^2;
eps4max^2;
eps5max^2
]));

Deps = zeros(length(Ae), p);
barR = Deps.' * Q * Deps + R;

Am = Ae + alpha * eye(size(Ae));
Bm = B1e;
Qm = Ceps.' * Q * Ceps;
Rm = barR;
Sm = (Deps.' * Q * Ceps).';
Em = eye(size(Ae));
Gm = 0;

[Xm, Km, Lm] = icare(Am, Bm, Qm, Rm, Sm, Em, Gm);
```

```

K    = -Km;
Ks   = K(1:length(Amatrix));
KIA  = K(length(Amatrix)+1:end);
```

Listing 1: LQR state-feedback controller with integral action

4.2 State Observer Design via Kalman Filter

Since not all states of the LKAS model are directly measurable from the camera, a state observer is designed to reconstruct the full state vector using the available output measurements. A Kalman filter–based observer is adopted, as it provides optimal state estimation in the presence of disturbances and measurement noise.

The observer is designed for the linearized system

$$\dot{x} = Ax + B_1 u + B_2 w, \qquad y = Cx + D_2 w,$$

where the vector w includes disturbances and sensor noise.

Disturbance and Noise Modeling

The uncertainty affecting the system is modeled through a covariance matrix Q_d, which captures the effect of disturbances and sensor noise. In particular, variations in the front and rear cornering stiffness are modeled as:

$$\sigma_{C_f} = 0.1\, C_f, \qquad \sigma_{C_r} = 0.1\, C_r,$$

leading to the disturbance covariance

$$Q_{d11} = \begin{bmatrix} \sigma_{C_f}^2 & 0 \\ 0 & \sigma_{C_r}^2 \end{bmatrix}.$$

The camera measurement noise is modeled as zero-mean white noise with variance

$$\sigma_\nu^2,$$

resulting in the measurement noise covariance $Q_{d22} = \sigma_\nu^2$. Cross-correlation terms between disturbances and noise are neglected, yielding the complete covariance matrix:

$$Q_d = \begin{bmatrix} Q_{d11} & 0 & 0 \\ 0 & Q_{d22} & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The measurement noise covariance matrix is defined as

$$R_d = \epsilon I,$$

where ϵ is a small positive scalar reflecting sensor accuracy.

Dual LQR Formulation for Observer Gain

The Kalman observer gain is computed using the duality between optimal control and optimal estimation. To improve numerical conditioning, a stability margin α_d is introduced by shifting the transposed system matrix:

$$A_m = A^\top + \alpha_d I, \qquad B_m = C^\top.$$

The equivalent cost matrices for the estimation problem are defined as:

$$Q_m = B_2 Q_d B_2^\top, \qquad R_m = D_2 Q_d D_2^\top + R_d, \qquad S_m = (D_2 Q_d B_2^\top)^\top.$$

The observer Riccati equation is solved using the `icare` algorithm, yielding the optimal estimation gain:

$$K_o = K_m^\top.$$

Observer Dynamics

The resulting observer reconstructs the state vector according to:

$$\dot{\hat{x}} = A\hat{x} + B_1 u + K_o(y - \hat{y}), \qquad \hat{y} = C\hat{x}.$$

This Kalman filter–based observer provides optimal state estimation by balancing model confidence and measurement reliability. It enables accurate reconstruction of lateral velocity, yaw rate, and heading angle using only camera-based lateral displacement measurements, making it well suited for LKAS applications.
The Kalman filter–based observer is implemented in MATLAB using a dual Riccati formulation. The complete implementation, including disturbance and noise modeling, is reported in Listing 2.

```
%% Observer via Kalman Filter

alphad = 1;                        % observer stability margin

% Disturbance covariance (cornering stiffness uncertainty)
sigma_C_f = 0.1 * C_fs;
sigma_C_r = 0.1 * C_fs;

Qd11 = diag([sigma_C_f^2, sigma_C_r^2]);

% Measurement noise covariance (camera)
sigma_nu = 1e-3;
Qd22 = sigma_nu^2;

% No cross-correlation assumed
Qd12 = zeros(2,1);

Qd = [ Qd11   Qd12   zeros(2,1);
Qd12.' Qd22   zeros(1);
zeros(1,4) ];

% Measurement noise matrix
```

```
Rd = 1e-3 * eye(q);

% Effective measurement noise
barRd = D2matrix * Qd * D2matrix.' + Rd;

% Dual system formulation
Am = Amatrix.' + alphad * eye(size(Amatrix));
Bm = Cmatrix.';

Qm = B2matrix * Qd * B2matrix.';
Rm = barRd;
Sm = (D2matrix * Qd * B2matrix.').';
Em = eye(size(Amatrix));
Gm = 0;

% Riccati equation for Kalman filter
[Xm0, Km0, Lm0] = icare(Am, Bm, Qm, Rm, Sm, Em, Gm);

% Observer gain
Ko = Km0.';
```

Listing 2: Kalman filter–based state observer implementation

The observer gain K_o balances model confidence and camera measurement reliability, enabling accurate reconstruction of lateral velocity, yaw rate,and heading angle using only lateral displacement measurements.

5 Conclusion and Future Improvements

5.1 Conclusion

The Lane Keeping Assist System (LKAS) developed in this work is based on a linearized two-degree-of-freedom bicycle model and relies exclusively on camera-based measurements of lateral displacement. A state-feedback control strategy using a Linear Quadratic Regulator (LQR) with integral action was designed to ensure accurate lane tracking and zero steady-state error. Since not all system states are directly measurable, a Kalman filter–based observer was implemented to reconstruct the full state vector from limited sensor information.

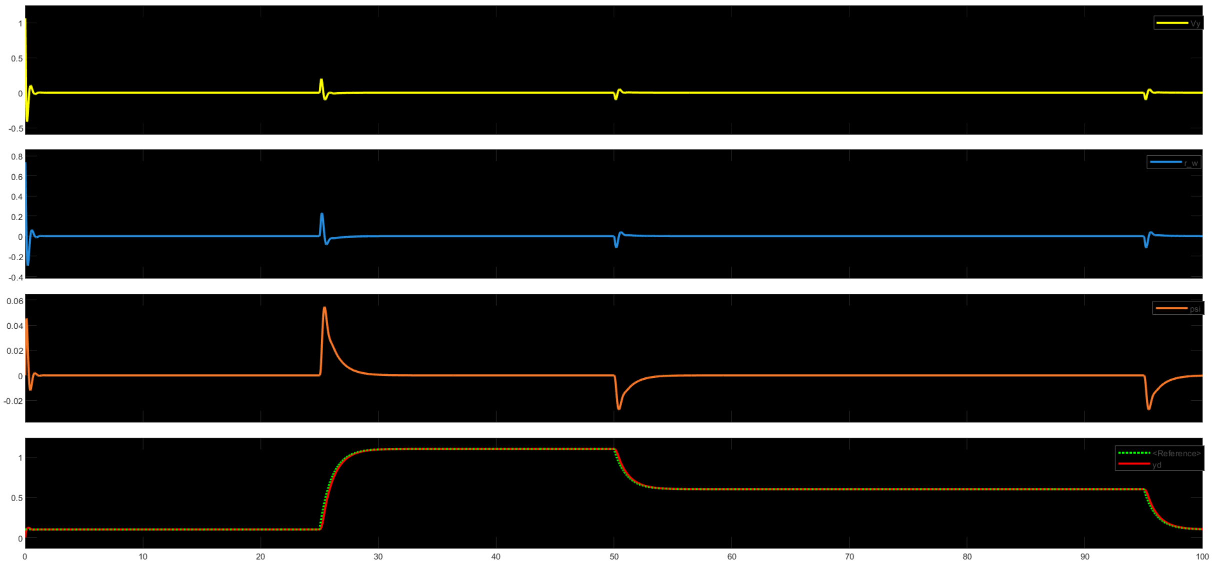

Figure 4: Non-Linear State Controlled

Simulation results confirm the effectiveness of the proposed approach. As shown in Fig. 4, the nonlinear closed-loop system maintains stable and accurate lane tracking under reference path variations. The lateral displacement y_d closely follows the reference with negligible steady-state error, validating the role of integral action. The remaining states—lateral velocity v_y, yaw rate r_ω, and heading angle ψ—remain bounded and well damped, indicating good closed-loop stability and passenger comfort.

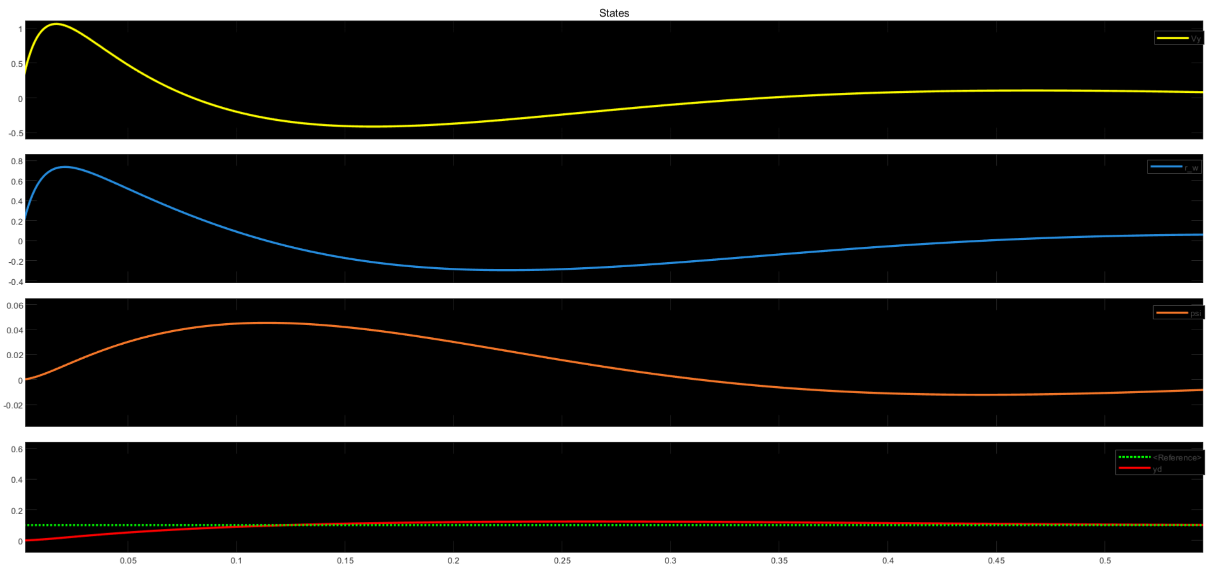

Figure 5: Initial State Controlled

The response to non-equilibrium initial conditions, illustrated in Fig. 5, demonstrates that the controller–observer structure rapidly stabilizes the vehicle and ensures convergence to the desired lane position. The observer performance confirms the observability analysis, showing that accurate state estimation can be achieved using only camera-based lateral displacement measurements. Overall, the results demonstrate that the proposed LKAS architecture provides a robust and effective solution for vision-based lateral vehicle control.

5.2 Future Improvements

Several extensions can further enhance the proposed LKAS framework. First, a feedforward steering term based on road curvature could be introduced to reduce transient tracking errors during lane curvature changes, improving anticipation capability. Second, the vehicle model could be extended to include roll dynamics, variable longitudinal speed, and steering actuator dynamics, increasing model fidelity under aggressive maneuvers. Additionally, incorporating input and state constraints would allow the use of constrained control strategies such as Model Predictive Control (MPC). Finally, experimental validation using real vehicle data or hardware-in-the-loop (HIL) testing would be a crucial next step to assess robustness and performance under real-world driving conditions.

References

References

[1] E. Ahmad, J. Iqbal, M. A. Khan, W. Liang, and I. Youn, "Predictive Control Using Active Aerodynamic Surfaces to Improve Ride Quality of a Vehicle," *Electronics*, vol. 9, no. 9, p. 1463, 2020. https://doi.org/10.3390/electronics9091463

[2] N. Mimmo, *Analysis and Design of Control Laws for Advanced Driver-Assistance Systems: Theory and Applications*. Cham, Switzerland: Springer, 2024. https://doi.org/10.1007/978-3-031-22520-8

[3] R. Rajamani, *Vehicle Dynamics and Control*, 2nd ed., New York, NY, USA: Springer, 2012. https://doi.org/10.1007/978-1-4614-1433-9