Longitudinal Active Suspension Control in a Half-Car Model with Unsprang Masses

Automatic Control
Electronic Engineering for Intelligent Vehicles
University of Bologna

A.A. 2025-2026

Alessandro Briccoli, Cristian Cecchini, Mario Di Marino

February 15, 2026

Abstract

This report presents the modeling, design, and simulation of an active suspension control system aimed at improving ride comfort and handling performance in passenger vehicles. The study focuses on a half-car model that includes both front and rear suspension dynamics, allowing the analysis of vertical and pitch motion of the vehicle body. Unlike passive suspension systems, which rely solely on spring-damper elements, the active suspension system introduced here incorporates actuators capable of generating controlled forces to counteract road disturbances in real time.

To achieve the desired ride quality, a control strategy based on Proportional-Integral-Derivative (PID) controllers is developed. Additionally, a state estimation technique using a Kalman observer is incorporated to provide real-time estimates of the states, helping to adapt to acceleration and road profile in a faster and more effective way.

The performance of the control system is evaluated through simulations conducted in MATLAB/Simulink. Results show a significant reduction in body acceleration and pitch angle variation when compared to a passive suspension system, demonstrating the effectiveness of the proposed approach in enhancing ride comfort. This project perfectly demonstrates, once again, the massive importance of active control strategies in automotive suspension design.

Chapter 1

Introduction

1.1 Motivations

Suspension systems play a crucial role in ensuring vehicle stability, comfort, and safety. Traditional passive suspensions cannot adapt to changing road conditions, leading to undesired oscillations and reduced performance.

In this project, focus on active suspension control in the longitudinal direction (front and rear), which aims to reduce vertical oscillations and pitch movements when the vehicle passes over bumps or uneven surfaces. The goal is to design a control system that keeps the car body as steady as possible, improving both passenger comfort and vehicle handling.

The half-car model was chosen because it provides a good trade-off between model simplicity and dynamic realism and allows to study the impact of front and rear suspension forces on the vehicle behavior without dealing with the complexity of a full 3D model.

1.2 List of the symbols

Table 1.1: Symbol Table

Symbol	Description	Dimension
x	State vector	$\mathbb{R}^{12}$
u	Control input vector $[u_1\ u_2]^T$	$\mathbb{R}^2$
y	Measured output vector	$\mathbb{R}^5$
e	Control error vector	$\mathbb{R}^2$
d	Disturbance vector	$\mathbb{R}^6$
r	Reference vector $[r_z\ r_\theta]^T$	$\mathbb{R}^2$
ν	Sensor noise vector	$\mathbb{R}^5$
w	Exogenous input $w = \text{col}(d, \nu, r)$	—
p_z	Vertical displacement of vehicle CoM	m
v_z	Vertical velocity of vehicle CoM	m/s
θ	Pitch angle of vehicle body	rad
$\dot{\theta}$	Pitch angular velocity	rad/s
h_{wf}, h_{wr}	Front and rear wheel vertical displacement	m
$\dot{h}_{wf}, \dot{h}_{wr}$	Front and rear wheel vertical velocity	m/s
θ_{gf}, θ_{gr}	Road pitch angle at front and rear axle	rad
z_{gf}, z_{gr}	Road vertical displacement at front and rear axle	m
u_1	Total vertical actuator force	N
u_2	Pitch moment component of actuator input	Nm
f_{af}, f_{ar}	Front and rear actuator forces	N
d_f, d_r	Distance from CoM to front and rear axle	m
h_{cg}	Height of vehicle center of mass	m
m	Sprung mass (vehicle body)	kg
m_{wf}, m_{wr}	Front and rear unsprung mass	kg
J	Pitch moment of inertia	$\text{kg}\,\text{m}^2$
k_f, k_r	Front and rear suspension stiffness	N/m
β_f, β_r	Front and rear suspension damping	N s/m
k_{tf}, k_{tr}	Front and rear tire stiffness	N/m
s_1, s_3	Front and rear suspension deflection	m
s_2, s_4	Front and rear suspension velocity	m/s
f_{sf}, f_{sr}	Front and rear suspension forces	N
f_{wf}, f_{wr}	Front and rear wheel dynamics forces	N
f_{xf}, f_{xr}	Longitudinal forces at front and rear axle	N
F_x	Total longitudinal force $F_x = f_{xf} + f_{xr}$	N
f_2	Vertical acceleration of vehicle body	m/s^2
f_4	Pitch angular acceleration	rad/s^2
α_{gf}, α_{gr}	Road pitch angular acceleration (front, rear)	rad/s^2
$\dot{z}_{gf}, \dot{z}_{gr}$	Road vertical velocity (front, rear)	m/s
y_x	Longitudinal accelerometer output	m/s^2
y_z	Vertical accelerometer output	m/s^2
y_θ	Gyroscope pitch-rate measurement	rad/s
y_f, y_r	Front and rear suspension sensors	m
$\nu_y, \nu_z, \nu_g, \nu_f, \nu_r$	Sensor noise components	—
r_z	Reference vertical position	m
r_θ	Reference pitch angle	rad
g	Gravitational acceleration	m/s^2
γ_f	Anti-Dive suspension inclination	—
γ_r	Anti-Squat suspension inclination	—

1.2.1 Dynamic Model

The considered vehicle model is a nonlinear longitudinal half-car representation equipped with two independently actuated suspensions. The model captures the vertical and pitch dynamics of the sprung mass, as well as the vertical dynamics of the front and rear unsprung masses. Tire compliance is explicitly included through linear tire stiffness, allowing the interaction between the wheels and the road profile to be accurately represented.

The sprung mass is assumed to be a rigid body with two degrees of freedom: vertical translation and pitch rotation. Each unsprung mass is modeled as a single vertical degree of freedom. Road excitations are treated as external inputs acting at the tire–road contact points and are not included as dynamic states.

State vector

The system state vector is defined as:

$$x = \begin{bmatrix} z_s & \dot{z}_s & \theta & \dot{\theta} & z_{wf} & \dot{z}_{wf} & z_{wr} & \dot{z}_{wr} \end{bmatrix}^T \in \mathbb{R}^8 \tag{1.1}$$

where z_s and $\dot{z}_s$ denote the vertical displacement and velocity of the sprung mass center of mass, θ and $\dot{\theta}$ are the pitch angle and pitch rate, while z_{wf}, z_{wr} and their time derivatives describe the vertical motion of the front and rear unsprung masses.

Control inputs

The active suspension system is driven by two control inputs:

$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{1.2}$$

where u_1 represents the total vertical force applied to the sprung mass, and u_2 is the pitching moment about the center of mass. The corresponding actuator forces at the front and rear suspensions are obtained through the static force distribution:

$$f_{af} = \frac{d_r u_1 + u_2}{d_f + d_r}, \tag{1.3}$$

$$f_{ar} = \frac{d_f u_1 - u_2}{d_f + d_r}, \tag{1.4}$$

with d_f and d_r denoting the distances from the center of mass to the front and rear axles, respectively.

Suspension kinematics

The suspension deflections and relative velocities are given by:

$$s_1 = z_s + d_f \sin\theta - z_{wf}, \qquad s_3 = z_s - d_r \sin\theta - z_{wr}, \tag{1.5}$$

$$s_2 = \dot{z}_s + d_f \dot{\theta}\cos\theta - \dot{z}_{wf}, \qquad s_4 = \dot{z}_s - d_r \dot{\theta}\cos\theta - \dot{z}_{wr}. \tag{1.6}$$

These expressions account for both the translational motion of the sprung mass and the geometric contribution due to pitch rotation.

Suspension and tire forces

The suspension forces are modeled as linear spring–damper elements:

$$f_{sf} = -k_f s_1 - \beta_f s_2, \tag{1.7}$$

$$f_{sr} = -k_r s_3 - \beta_r s_4, \tag{1.8}$$

where k_f, k_r and β_f, β_r are the stiffness and damping coefficients of the front and rear suspensions.

The tire–road interaction is described using linear tire stiffness:

$$f_{tf} = k_{tf}(z_{rf} - z_{wf}), \tag{1.9}$$

$$f_{tr} = k_{tr}(z_{rr} - z_{wr}), \tag{1.10}$$

with z_{rf} and z_{rr} denoting the vertical road displacements at the front and rear contact points.

Sprung mass dynamics

The vertical acceleration of the sprung mass follows from Newton's second law:

$$\ddot{z}_s = -g + \frac{1}{m}\left(f_{sf} + f_{sr} + f_{af} + f_{ar}\right). \tag{1.11}$$

The pitch dynamics about the center of mass are governed by:

$$\ddot{\theta} = \frac{1}{J}\left(d_f(f_{sf} + f_{af}) - d_r(f_{sr} + f_{ar}) + h_{cg}F_x\right), \tag{1.12}$$

where J is the pitch moment of inertia, h_{cg} is the height of the center of mass, and

$$F_x = F_{xf} + F_{xr} \tag{1.13}$$

is the resultant longitudinal force acting on the vehicle. This term accounts for the coupling between longitudinal forces and pitch dynamics through load transfer effects.

Unsprung mass dynamics

The vertical dynamics of the front and rear unsprung masses are given by:

$$\ddot{z}_{wf} = \frac{1}{m_{wf}} \left(f_{tf} - f_{sf} - f_{af} - m_{wf} g \right), \tag{1.14}$$

$$\ddot{z}_{wr} = \frac{1}{m_{wr}} \left(f_{tr} - f_{sr} - f_{ar} - m_{wr} g \right). \tag{1.15}$$

State-space representation

Collecting all terms, the nonlinear vehicle dynamics can be written in first-order state-space form as:

$$\dot{x} = f(x, u, w, r), \tag{1.16}$$

where w represents external longitudinal force disturbances and r collects the road profile inputs at the wheel contact points.

This formulation provides a complete nonlinear description of the vertical and pitch dynamics of the vehicle, forming the basis for linearization, controller synthesis, and observer design.

1.2.2 Sensor Model

The control architecture relies on a set of onboard sensors that provide real-time measurements of the vehicle vertical and pitch dynamics. The selected sensor suite is designed to ensure observability of the nonlinear half-car model and to provide sufficient information for feedback control in the presence of road disturbances and measurement noise.

All sensor measurements are assumed to be affected by additive noise and are expressed in the vehicle body-fixed reference frame.

Measurement Vector

The complete measurement vector is defined as:

$$y = \begin{bmatrix} y_y \\ y_z \\ y_g \\ y_f \\ y_r \end{bmatrix} = \begin{bmatrix} \sin(\theta)(f_2 + g) + \cos(\theta)\frac{F_x}{m} \\ \cos(\theta)(f_2 + g) - \sin(\theta)\frac{F_x}{m} \\ \dot{\theta} \\ s_1 \\ s_3 \end{bmatrix} + \begin{bmatrix} \nu_y \\ \nu_z \\ \nu_g \\ \nu_f \\ \nu_r \end{bmatrix}. \tag{1.17}$$

This vector includes measurements from two accelerometers, one gyroscope, and two suspension displacement sensors.

Accelerometer Model

The first two outputs y_y and y_z represent the longitudinal and vertical accelerations measured in the body-fixed reference frame. These signals are obtained from MEMS accelerometers mounted near the vehicle center of mass.

The accelerometer outputs are nonlinear functions of the vehicle dynamics and include both inertial and gravitational contributions. The term $f_2 + g$ corresponds to the absolute vertical acceleration of the sprung mass, while F_x/m represents the longitudinal acceleration induced by external forces.

The trigonometric terms $\sin(\theta)$ and $\cos(\theta)$ account for the projection of the acceleration vector onto the body-fixed axes due to the pitch angle, resulting in an intrinsic coupling between vertical and pitch dynamics.

A typical automotive-grade device is the **STMicroelectronics LIS3DH**, providing 12-bit resolution and low noise density.

Gyroscope Model

The third measurement y_g corresponds to the pitch angular rate:

$$y_g = \dot{\theta} + \nu_g. \tag{1.18}$$

This signal is provided by a gyroscope rigidly attached to the vehicle body and delivers high-bandwidth information essential for pitch stabilization and transient response improvement.

Figure 1.1: MEMS accelerometer LIS3DH

A representative sensor is the **Bosch BMI160**, integrating accelerometer and gyroscope within a compact IMU package.

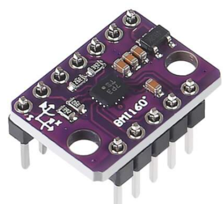


Figure 1.2: Bosch BMI160 IMU

Suspension Deflection Sensors

The last two measurements y_f and y_r correspond to the front and rear suspension deflections:

$$y_f = s_1 + \nu_f, \qquad y_r = s_3 + \nu_r. \tag{1.19}$$

The deflections are defined as:

$$s_1 = (z_s + d_f \sin\theta) - z_{wf}, \tag{1.20}$$

$$s_3 = (z_s - d_r \sin\theta) - z_{wr}, \tag{1.21}$$

and represent the relative displacement between the sprung mass and the unsprung masses. These measurements provide direct information on road excitation and wheel–body interaction.

Linear potentiometers such as the **SLS190** are assumed.

Figure 1.3: Linear suspension potentiometer

Noise Modeling

All sensor measurements are affected by additive noise:

$$\nu = \begin{bmatrix} \nu_y & \nu_z & \nu_g & \nu_f & \nu_r \end{bmatrix}^T . \tag{1.22}$$

Noise components are modeled as zero-mean Gaussian processes with variances derived from typical automotive sensor specifications. Accelerometers and gyroscopes include white noise and bias drift, while suspension sensors are mainly affected by quantization and thermal noise.

1.3 Performance Variables and Control Objectives

To evaluate the effectiveness of the active suspension system and to formulate the optimization problem (e.g., for optimal control or performance monitoring), a specific performance output vector $z \in \mathbb{R}^6$ is defined. Unlike the measured output vector y, which depends on available sensors, the vector z contains the physical variables that strictly define the ride quality and the mechanical constraints of the vehicle.

The performance vector is constructed as a function of the state vector x, the vehicle parameters p, and the reference vector r:

$$z = g(x, p, r) \tag{1.23}$$

Control Objectives

The control objective is defined in terms of vertical position and perceived pitch regulation. An apparent pitch angle θ_a is reconstructed from accelerometer data as:

$$\theta_a = \sin^{-1}\left(\frac{y_y}{\sqrt{y_y^2 + y_z^2}}\right) . \tag{1.24}$$

The control error vector is defined as:

$$e = \begin{bmatrix} \dfrac{(s_1+\nu_f)d_r + (s_3+\nu_r)d_f}{d_r+d_f} - r_z \\ \theta_a - r_\theta \end{bmatrix}, \tag{1.25}$$

where r_z and r_θ denote the desired vertical position and pitch angle references. The first component regulates the vehicle vertical displacement through a weighted average of suspension deflections, while the second penalizes deviations from the perceived pitch angle experienced by passengers.

Output Error Function

The complete nonlinear output error function used for control design is:

$$he(x,u,w) = \begin{bmatrix} \dfrac{(s_1+\nu_f)d_r + (s_3+\nu_r)d_f}{d_r+d_f} - r_z \\ \sin^{-1}\left(\dfrac{h_1+\nu_y}{\sqrt{(h_1+\nu_y)^2+(h_2+\nu_z)^2}}\right) - r_\theta \end{bmatrix}, \tag{1.26}$$

where h_1 and h_2 represent the ideal accelerometer outputs derived from the system dynamics.

Observability Considerations

The selected sensor configuration ensures observability of the states relevant for control. Accelerometer and gyroscope measurements capture translational and rotational dynamics, while suspension deflection sensors resolve wheel–body interaction and road-induced disturbances, enabling reliable state estimation through standard observers.

Augmented Performance Output for Control Design

In addition to the nonlinear output error function $h_e(x,u,w)$ previously defined for tracking purposes, the performance output vector has been extended to match the complete Simulink control architecture and to explicitly account for dynamic and structural constraints of the half-car model.

Specifically, the performance vector is no longer limited to static regulation of vertical position and perceived pitch angle, but it also includes additional dynamic quantities required to improve transient behavior, damping characteristics, and suspension stroke monitoring.

The resulting performance vector is defined as:

$$z = g(p,x,r) \in \mathbb{R}^6, \tag{1.27}$$

with the following structure:

$$z = \begin{bmatrix} p_z - r_z \\ \theta - r_\theta \\ \omega \\ s_f \\ s_r \\ v_z \end{bmatrix}. \tag{1.28}$$

Definition of Suspension Deflections The suspension deflections are explicitly reconstructed from the kinematic relations of the half-car model as:

$$s_f = (p_z + d_f \sin\theta) - z_{wf}, \tag{1.29}$$

$$s_r = (p_z - d_r \sin\theta) - z_{wr}, \tag{1.30}$$

where:

- p_z is the vertical position of the center of gravity;
- θ is the pitch angle;
- z_{wf} and z_{wr} are the front and rear wheel hub vertical positions;
- d_f and d_r are the longitudinal distances between the center of gravity and the front and rear axles.

Physical Meaning of the Augmented Components The augmentation of the performance vector introduces four additional quantities beyond the basic tracking errors:

1. ω (pitch rate): This term is included to provide additional rotational damping. Penalizing ω improves transient response and reduces oscillatory pitch dynamics.
2. s_f and s_r (front and rear suspension deflections): These terms allow the controller to indirectly monitor suspension stroke limits (jounce and rebound). Their inclusion prevents excessive suspension travel and improves ride comfort and safety.
3. v_z (vertical velocity of the center of gravity): This component contributes to heave damping, reducing vertical oscillations and improving passenger comfort.

Relation with the Output Error Function The first two components of z are consistent with the regulation objectives previously introduced through the nonlinear output error function $h_e(x, u, w)$. In particular:

- The vertical position error corresponds to the regulation of the vehicle heave motion.

- The pitch error enforces alignment with the desired pitch reference.

The additional components do not modify the reference definitions r_z and r_θ, and no previously introduced variable names have been altered. Instead, the performance matrix has been augmented to reflect the complete Simulink block implementation, where tracking, damping, and structural constraints are treated simultaneously within the control synthesis framework.

Control-Oriented Interpretation From a control design perspective, the augmented vector z defines the full set of regulated outputs used for performance shaping. This formulation allows:

- reference tracking (heave and pitch);
- oscillation damping (through v_z and ω);
- suspension constraint management (through s_f and s_r).

1.3.1 System Linearization

To facilitate the control design of the longitudinal half-car model equipped with active front and rear suspension systems, the initial step involves the linearization of the nonlinear system dynamics. This process is performed by identifying appropriate steady-state operating points (x^*, y^*, w^*), which characterize representative conditions under which the vehicle is expected to operate. Linearizing the system around these equilibrium points enables the derivation of a time-invariant linear approximation of the vehicle dynamics, thereby simplifying the synthesis and analysis of control strategies.

Linearization Around the Operating Point

Consider the nonlinear system model:

$$\begin{aligned} \dot{x} &= f(x, u, w), \quad x(t_0) = x_0 \\ y &= h(x, u, w) \\ e &= h_e(x, u, w) \end{aligned} \tag{1.31}$$

The steady-state operating points $(x^\star, u^\star, w^\star)$ is called *equilibrium triplet* if satisfies the condition:

$$f(x^\star, u^\star, w^\star) = 0 \tag{1.32}$$

and defines the equilibrium output and error as:

$$y^\star := h(x^\star, u^\star, w^\star), \quad e^\star := h_e(x^\star, u^\star, w^\star) \tag{1.33}$$

The variations around the equilibrium point are defined as:

$$\begin{aligned} \tilde{x} &:= x - x^\star \\ \tilde{y} &:= y - y^\star \\ \tilde{e} &:= e - e^\star \\ \tilde{u} &:= u - u^\star \\ \tilde{w} &:= w - w^\star \end{aligned} \tag{1.34}$$

Using the fact that $\dot{x}^\star = 0$, the dynamics of the variations are:

$$\begin{aligned} \dot{\tilde{x}} &= f(x^\star + \tilde{x}, u^\star + \tilde{u}, w^\star + \tilde{w}), \quad \tilde{x}(t_0) = x_0 - x^\star \\ \tilde{y} &= h(x^\star + \tilde{x}, u^\star + \tilde{u}, w^\star + \tilde{w}) \\ \tilde{e} &= h_e(x^\star + \tilde{x}, u^\star + \tilde{u}, w^\star + \tilde{w}) \end{aligned} \tag{1.35}$$

To obtain a tractable model for controller synthesis, a first-order Taylor expansion is applied around the equilibrium point. The resulting Jacobian matrices are defined as:

$$
\begin{aligned}
A &:= \left.\frac{\partial f(x,u,w)}{\partial x}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}} & B_1 &:= \left.\frac{\partial f(x,u,w)}{\partial u}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}} & B_2 &:= \left.\frac{\partial f(x,u,w)}{\partial w}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}}\\
C &:= \left.\frac{\partial h(x,u,w)}{\partial x}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}} & D_1 &:= \left.\frac{\partial h(x,u,w)}{\partial u}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}} & D_2 &:= \left.\frac{\partial h(x,u,w)}{\partial w}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}}\\
C_e &:= \left.\frac{\partial h_e(x,u,w)}{\partial x}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}} & D_{e1} &:= \left.\frac{\partial h_e(x,u,w)}{\partial u}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}} & D_{e2} &:= \left.\frac{\partial h_e(x,u,w)}{\partial w}\right|_{\substack{x=x^\star\\u=u^\star\\w=w^\star}}
\end{aligned}
\tag{1.36}
$$

Neglecting second-order terms, the linearized system becomes the so-called *design model*:

$$
\begin{cases}
\dot{\tilde{x}} = A\tilde{x} + B_1\tilde{u} + B_2\tilde{w}, \quad \tilde{x}(t_0) = x_0 - x^\star\\
\tilde{y} = C\tilde{x} + D_1\tilde{u} + D_2\tilde{w}\\
\tilde{e} = C_e\tilde{x} + D_{e1}\tilde{u} + D_{e2}\tilde{w}
\end{cases}
\tag{1.37}
$$

This Linear Time-Invariant (LTI) approximation of the nonlinear model is valid in a neighborhood of the equilibrium point, enabling efficient analysis and controller design under small perturbations.

Matrix calculus

For the matrix calculation, the procedure described in equations (1.36) was followed. By substituting the equilibrium triplet given in (1.32), obtaining the following matrices:

$$
A = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-\frac{k_f+k_r}{m} & -\frac{\beta_f+\beta_r}{m} & -\frac{d_fk_f-d_rk_r}{m} & -\frac{\beta_fd_f-\beta_rd_r}{m} & \frac{k_f}{m} & \frac{\beta_f}{m} & \frac{k_r}{m} & \frac{\beta_r}{m} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
-\frac{d_fk_f-d_rk_r}{J} & -\frac{\beta_fd_f-\beta_rd_r}{J} & -\frac{k_fd_f^2+k_rd_r^2}{J} & -\frac{\beta_fd_f^2+\beta_rd_r^2}{J} & \frac{d_fk_f}{J} & \frac{\beta_fd_f}{J} & -\frac{d_rk_r}{J} & -\frac{\beta_rd_r}{J} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
\frac{k_f}{m_{wf}} & \frac{\beta_f}{m_{wf}} & \frac{d_fk_f}{m_{wf}} & \frac{\beta_fd_f}{m_{wf}} & -\frac{k_f+k_{tf}}{m_{wf}} & -\frac{\beta_f}{m_{wf}} & 0 & 0 & 0 & 0 & \frac{k_{tf}}{m_{wf}} & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\
\frac{k_r}{m_{wr}} & \frac{\beta_r}{m_{wr}} & -\frac{d_rk_r}{m_{wr}} & -\frac{\beta_rd_r}{m_{wr}} & 0 & 0 & -\frac{k_r+k_{tr}}{m_{wr}} & -\frac{\beta_r}{m_{wr}} & 0 & 0 & 0 & \frac{k_{tr}}{m_{wr}}\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\tag{1.38}
$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & \frac{1}{J} \\ 0 & 0 \\ -\frac{d_r}{m_{wf}(d_f+d_r)} & -\frac{1}{m_{wf}(d_f+d_r)} \\ 0 & 0 \\ -\frac{d_f}{m_{wr}(d_f+d_r)} & \frac{1}{m_{wr}(d_f+d_r)} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{1.39}$$

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{h_{cg}}{J} & \frac{h_{cg}}{J} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.40}$$

$$C = \begin{bmatrix} 0 & 0 & \frac{0.350357(k_f+k_r)}{m} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{k_f+k_r}{m} & -\frac{\beta_f+\beta_r}{m} & -\frac{d_f k_f - d_r k_r}{m} & -\frac{\beta_f d_f - \beta_r d_r}{m} & \frac{k_f}{m} & \frac{\beta_f}{m} & \frac{k_r}{m} & \frac{\beta_r}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & d_f & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -d_r & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.41}$$

$$D_1 = \begin{bmatrix} 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{1.42}$$

$$
D_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{m} & \frac{1}{m} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{1.43}
$$

$$
D_{e1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{1.44}
$$

$$
D_{e2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{d_r}{d_f + d_r} & \frac{d_f}{d_f + d_r} & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{1.45}
$$

$$
C_e = \begin{bmatrix} 1 & 0 & 0 & 0 & -\frac{d_r}{d_f + d_r} & 0 & -\frac{d_f}{d_f + d_r} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.46}
$$

Numerical Matrices To calculate the numerical values of the matrices, the parameters were replaced with the values of the vehicle chosen for this study: All parameters used to find the numeric matrices are based on the Tesla Model S, a car suitable for this type of active suspensions and with same suspension paramethers on front and rear axle, this simplifies slighlty our model:

Figure 1.4: Rolls-Royce Ghost 6.7 V12

Symbol	Description	Value	Unit
m	Total mass of the vehicle	2550	kg
g	Gravitational acceleration	9.81	m/s^2
h_{cg}	Height of center of gravity	0.60	m
d_f	Front axle distance from CoM	1.65	m
d_r	Rear axle distance from CoM	1.65	m
L	Wheelbase	3.30	m
J	Pitch moment of inertia	4080	kg·m^2
k_f	Front suspension stiffness	35000	N/m
k_r	Rear suspension stiffness	32000	N/m
β_f	Front damping coefficient	4200	N·s/m
β_r	Rear damping coefficient	4200	N·s/m
m_{wf}	Front unsprung mass	48	kg
m_{wr}	Rear unsprung mass	48	kg
k_{tf}	Front tire stiffness	270000	N/m
k_{tr}	Rear tire stiffness	270000	N/m
ℓ_0	Static suspension height	0.55	m

Table 1.2: Dynamic model parameters of a Rolls-Royce vehicle (Half-Car Model)

$$A = 10^3 \begin{bmatrix} 0 & 0.0010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0280 & -0.0036 & -0.0009 & 0.0009 & 0.0164 & 0.0018 & 0.0116 & 0.0018 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0010 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0004 & 0.0004 & -0.0340 & -0.0046 & 0.0105 & 0.0012 & -0.0101 & -0.0016 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0010 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.9111 & 0.1000 & 1.2756 & 0.1400 & -7.1333 & -0.1000 & 0 & 0 & 0 & 0 & 6.2222 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.0010 & 0 & 0 & 0 & 0 \\ 0.6444 & 0.1000 & -1.2244 & -0.1900 & 0 & 0 & -6.8667 & -0.1000 & 0 & 0 & 0 & 6.2222 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.47}$$

$$B_1 = \begin{bmatrix} 0 & 0 \\ 0.000921659 & 0 \\ 0 & 0 \\ 0 & 0.000205339 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{1.48}$$

$$B_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0000662428 & 0.0000662428 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{1.49}$$

$$C = \begin{bmatrix} 0 & 0 & 9.81 & 0 & 0 & 0 & 0 & 0 \\ -55.2995 & -3.68664 & 6.52535 & 0.435023 & 37.659 & -44.1843 & 2.5106 & -2.94562 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1.362 & 0 & -1.362 & 0 & 0 & 0 \\ 1 & 0 & -1.598 & 0 & 0 & 1.598 & 0 & 0 \end{bmatrix} \tag{1.50}$$

$$D_1 = \begin{bmatrix} 0 & 0 \\ 0.000921659 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{1.51}$$

$$D_2 = \begin{bmatrix} 0 & 0 & 0 & 0.000921659 & 0.000921659 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{1.52}$$

$$C_e = \begin{bmatrix} 1 & 0 & 0 & 0 & -0.735296 & 0.735296 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.53}$$

$$D_{e2} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.539865 & 0.460135 & -1 & 0 \\ 0 & 0 & 0 & 0.000093951 & 0.000093951 & 0.101937 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{1.54}$$

1.3.2 Linear Model Analysis

To support the development of an effective suspension controller and gain insight into the vehicle's dynamic behavior, we examine a linearized model of a half-car system. This analysis is performed around a nominal equilibrium point, where the vehicle is at rest with zero pitch angle and no vertical or angular velocities. The focus is on the open-loop dynamics of the vehicle body, excluding actuator behavior and wheel compliance.

The model captures the essential vertical (heave) and angular (pitch) motions of the chassis. The state vector is defined as:

$$x = \begin{bmatrix} z - z_0 \\ \dot{z} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{1.55}$$

where z_0 is the vertical position of the center of mass at static equilibrium, and θ denotes the pitch angle of the chassis. The state-space representation of the system is given by:

$$\dot{x} = A_{\text{int}} x \tag{1.56}$$

Analysis of this reduced 4x4 matrix will be performed, ignoring the 4 exogenous states, the reason behind this is explained later in chapter 1.4 (Reachability) **todo: ricalcolare**

$$A_{\text{int}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -55.2995 & -3.68664 & 6.52535 & 0.435023 \\ 0 & 0 & 0 & 1 \\ 1.4538 & 0.0969199 & -27.158 & -1.81053 \end{bmatrix} \tag{1.57}$$

This matrix characterizes the coupled dynamics of the system, with off-diagonal elements indicating interaction between vertical and angular motions. Such coupling arises naturally in a physical half-car system, where vertical displacements can induce pitching, and vice versa.

1.3.3 Open-Loop Dynamics

To evaluate the system's stability and transient behavior, is necessary to calculate the eigenvalues of the matrix A_{int} by solving the characteristic equation:

$$\det(A_{\text{int}} - \lambda I) = 0 \tag{1.58}$$

The resulting eigenvalues are:

todo: ricalcolare

$$\lambda_{1,2} = -1.55 \pm j6.28, \quad \lambda_{3,4} = -2.47 \pm j5.10 \tag{1.59}$$

The eigenvalues form two pairs of complex conjugates, each with negative real parts, indicating that the system is asymptotically stable in open loop. These pairs represent distinct oscillatory modes characterized by different damping and frequency properties:

- The eigenvalues $\lambda_{1,2}$ correspond to a lightly damped oscillatory mode with a higher frequency.
- The eigenvalues $\lambda_{3,4}$ correspond to a more heavily damped mode with a lower oscillation frequency.

Due to the coupling present in A_{int}, these modes represent mixed heave-pitch dynamics rather than purely vertical or angular motions. This hybrid behavior implies that any disturbance in one degree of freedom can propagate to the other, necessitating a control strategy that accounts for this interaction.

The system's open-loop stability ensures that disturbances decay over time, but the oscillatory nature of the transients may degrade ride comfort and vehicle handling. Therefore, active suspension control is warranted to suppress these vibrations more rapidly, reduce settling time, and improve overall vehicle performance.

In summary, the linearized model described by A_{int} reveals a stable but dynamically coupled system with oscillatory characteristics. These insights are essential for the design of coordinated control strategies that enhance both ride quality and dynamic response.

1.4 Control

Reachability

In this section, a reachability analysis is carried out to determine which parts of the system's state space can be influenced by the control input. Starting from the linearized state-space system:

$$\dot{\tilde{x}} = A\tilde{x} + B_1\tilde{u} \tag{1.60}$$

we aim to identify which states can be driven from the origin to a desired position through the control input $\tilde{u}$. The reachability matrix is defined as:

$$\mathcal{R} = \begin{bmatrix} B_1 & AB_1 & A^2B_1 & \dots & A^{n-1}B_1 \end{bmatrix} \tag{1.61}$$

A linear time-invariant system is said to be **fully reachable** (or controllable from the origin) if the rank of $\mathcal{R}$ is equal to n, the number of state variables. In such a case, it is possible to design a state feedback controller that places all eigenvalues of the closed-loop system arbitrarily.

In our case, the system has $n = 8$ state variables. These include both the vehicle body dynamics (vertical position and velocity, pitch angle and rate) and road-related exogenous components (front and rear road pitch angles and their derivatives). The control input $\tilde{u} \in \mathbb{R}^2$ acts through actuators located at the front and rear suspensions, influencing primarily the body dynamics of the vehicle.

However, the states related to the road excitation are not directly controllable. As such, only a subset of the full state vector can be affected by $\tilde{u}$. Upon computation of the reachability matrix $\mathcal{R}$ using the specific system matrices A and B_1, we obtain:
todo: ricalcolare

```
R = ctrb(A, B1);
rank_R = rank(R);
```

$$\text{rank}(\mathcal{R}) = 4 < 8 \tag{1.62}$$

This indicates that the system is **not fully reachable**. Only the four states corresponding to the internal dynamics of the vehicle body are reachable, while the remaining four states — associated with the road profile — are uncontrollable and must be treated as external disturbances. Consequently, control strategies such as state feedback and optimal control (e.g., LQR) can only be designed for the reachable subspace.

Reduced Reachability Analysis

In the present model, the state vector $\tilde{x} \in \mathbb{R}^8$ includes eight components. However, the last four states represent environmental or road-related variables (e.g., road profile curvature), which evolve independently of the control input $\tilde{u}$. These are known as **exogenous states** and are not directly controllable.

Therefore, reachability analysis is performed only on the first four states, which represent the internal vehicle dynamics and are influenced by control inputs. Let A_{int} and $B_{1,\text{int}}$ be the upper-left 4×4 and 4×2 blocks of matrices A and B_1:

todo: ricalcolare:

$$A_{\text{int}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -55.2995 & -3.68664 & 6.52535 & 0.435023 \\ 0 & 0 & 0 & 1 \\ 1.4538 & 0.0969199 & -27.158 & -1.81053 \end{bmatrix} \tag{1.63}$$

$$B_{1,\text{int}} = \begin{bmatrix} 0 & 0 \\ 0.000921659 & 0 \\ 0 & 0 \\ 0 & 0.000205339 \end{bmatrix} \tag{1.64}$$

$$\dot{\tilde{x}}_{\text{int}} = A_{\text{int}}\tilde{x}_{\text{int}} + B_{1,\text{int}}\tilde{u} \tag{1.65}$$

The reachability matrix becomes:

$$\mathcal{R}_{\text{int}} = \begin{bmatrix} B_{1,\text{int}} & A_{\text{int}}B_{1,\text{int}} & A_{\text{int}}^2 B_{1,\text{int}} & A_{\text{int}}^3 B_{1,\text{int}} \end{bmatrix} \tag{1.66}$$

An evaluation of the rank of this matrix using MATLAB's `ctrb` function:

```
R_int = ctrb(A_int, B1_int);
rank(R_int)
```

The resulting rank is 4, which confirms that the internal subsystem is fully reachable.

Stabilizability and the Hurwitz Condition According to control theory, if the system is fully reachable, it is possible to design a state feedback control law:

$$\tilde{u} = K_S \tilde{x}_{\text{int}} \tag{1.67}$$

such that the closed-loop matrix:

$$A_{\text{int}} + B_{1,\text{int}} K_S \tag{1.68}$$

is **Hurwitz**, meaning that all of its eigenvalues lie in the left half of the complex plane. This ensures that the closed-loop system is **BIBS stable** (Bounded

Input Bounded State): all state trajectories remain bounded in response to bounded inputs.

In conclusion, while the full model is not entirely reachable due to the presence of exogenous states, the subsystem representing the vehicle dynamics is fully reachable and can be stabilized using linear state feedback.

Integral Action

While the stabilizer matrix K_S ensures the stability of the internal system under feedback control, an integral action is required to eliminate steady-state error in the presence of unknown constant disturbances $\tilde{w}$.

The regulated error $\tilde{e}$ is defined as:

$$\tilde{e} = C_e\tilde{x} + D_{1e}\tilde{u} + D_{2e}\tilde{w} \tag{1.69}$$

To implement integral control, new integral state η is introduced:

$$\dot{\eta} = \tilde{e} \tag{1.70}$$

This leads to an extended state vector:

$$x_e = \begin{bmatrix} \tilde{x} \\ \eta \end{bmatrix} \tag{1.71}$$

The extended system dynamics are:

$$\dot{x}_e = \bar{A}x_e + \bar{B}_1\tilde{u} + \bar{B}_2\tilde{w} \tag{1.72}$$

where

$$\bar{A} = \begin{bmatrix} A & 0 \\ C_e & 0 \end{bmatrix}, \quad \bar{B}_1 = \begin{bmatrix} B_1 \\ D_{1e} \end{bmatrix}, \quad \bar{B}_2 = \begin{bmatrix} B_2 \\ D_{2e} \end{bmatrix} \tag{1.73}$$

To verify whether the extended system is controllable (reachable), a reachability matrix is computed as:

$$\mathcal{R}_e = \begin{bmatrix} \bar{B}_1 & \bar{A}\bar{B}_1 & \bar{A}^2\bar{B}_1 & \cdots & \bar{A}^{n+q-1}\bar{B}_1 \end{bmatrix} \tag{1.74}$$

where n is the number of original state variables and q is the number of integrator states. In this model, $n = 4$ (excluding the four uncontrollable states) and $q = 2$, so we expect:

$$\text{rank}(\mathcal{R}_e) = 6 \tag{1.75}$$

The reachability matrix is constructed in MATLAB using:

```
Ae = [A, zeros(6,2); Ce, zeros(2,2)];
B1e = [B1; D1e];
Re = ctrb(Ae, B1e);
rank(Re)
```

Result: The output of the command confirms that $\text{rank}(\mathcal{R}_e) = 6$, i.e., the extended system is fully reachable.

Conclusion: Since the extended system is reachable, it is possible to design a state feedback control law of the form:

$$\tilde{u} = \bar{K}x_e = K_S\tilde{x} + K_I\eta \tag{1.76}$$

such that the closed-loop matrix $\bar{A} + \bar{B}_1\bar{K}$ is Hurwitz. This guarantees asymptotic stability of the augmented system and drives the regulation error $\tilde{e}$ to zero.

Observability

Up to this point, the system state $\tilde{x}$ has been assumed to be known. However, this assumption is often unrealistic, particularly when some state variables are not directly measurable. For this reason, a state observer is required to estimate the internal states based on measurable outputs.

Given that the last four states of $\tilde{x}$ are exogenous and not influenced by the system dynamics, we restrict our observability analysis to the internal dynamics only, i.e., $\tilde{x}_{\text{int}} \in \mathbb{R}^4$.

Let C_{int} be the output matrix corresponding to the internal states. Then, the observability matrix is given by:

$$\mathcal{O} = \begin{bmatrix} C_{\text{int}} \\ C_{\text{int}}A_{\text{int}} \\ C_{\text{int}}A_{\text{int}}^2 \\ C_{\text{int}}A_{\text{int}}^3 \end{bmatrix} \tag{1.77}$$

$$C_{\text{int}} = \begin{bmatrix} 0 & 0 & 9.81 & 0 \\ -55.2995 & -3.68664 & 6.52535 & 0.435023 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1.362 & 0 \\ 1 & 0 & -1.598 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{1.78}$$

Using MATLAB, it is computed that:

```
O = obsv(A_int, C_int);
rank(O)
```

If $\text{rank}(\mathcal{O}) = n = 4$, the system is **fully observable**, and it is possible to construct a full-order **Kalman filter** (optimal observer):

$$\dot{\hat{x}}_{\text{int}} = A_{\text{int}}\hat{x}_{\text{int}} + B_{1,\text{int}}\tilde{u} + K_f(y - C_{\text{int}}\hat{x}_{\text{int}}) \tag{1.79}$$

where K_f is the **Kalman gain**, computed to optimally minimize the estimation error covariance under the assumption of white Gaussian process and measurement noise. The Kalman gain is derived from the steady-state solution of the continuous-time algebraic Riccati equation:

$$A_{\text{int}}P + PA_{\text{int}}^T - PC_{\text{int}}^T R^{-1} C_{\text{int}} P + Q = 0 \tag{1.80}$$

$$K_f = PC_{\text{int}}^T R^{-1} \tag{1.81}$$

Here:

- Q is the covariance matrix of the process noise,
- R is the covariance matrix of the measurement noise,
- P is the steady-state error covariance matrix.

By solving the Riccati equation with appropriately chosen Q and R, the filter gain K_f ensures that the estimation error converges in a statistically optimal sense, even in the presence of measurement and process disturbances.

Conclusion:
The MATLAB analysis confirms that $\text{rank}(\mathcal{O}) = 4$, which equals the number of internal states. Therefore, the pair $(A_{\text{int}}, C_{\text{int}})$ is **fully observable**, and a Kalman filter can be designed. This guarantees that despite partial measurements, the entire internal state vector $\tilde{x}_{\text{int}}$ can be optimally reconstructed for control and estimation purposes.

This observability ensures that despite partial measurements, the entire controllable state can be reconstructed and regulated accordingly.

Chapter 2

MATLAB/Simulink Implementation

2.0.1 Overall Simulation Architecture

The complete control framework has been implemented in MATLAB/Simulink using a modular structure that separates the nonlinear plant, the linearized model used for control synthesis, the state observer, the performance evaluation block, and the disturbance generators.

The simulation architecture is organized into the following main subsystems:

- Nonlinear Half-Car Plant (blocks f, h, and e),
- Road Profile Generator,
- Longitudinal Wheel Forces Generator,
- Linearized Model (for control design and validation),
- State Observer,
- Augmented LQR Controller with integral action,
- Enable logic and scopes for scenario-based simulations.

This structure ensures coherence between the analytical derivations, the symbolic linearization procedure, and the numerical simulations.

2.0.2 Symbolic Linearization and Control Synthesis Script

The core of the control-oriented implementation is the MATLAB script dedicated to symbolic linearization and numerical evaluation of the half-car model.

This file plays a fundamental role in the overall workflow, since it:

- Defines the full nonlinear model in symbolic form,

- Computes all Jacobian matrices required for linearization,
- Evaluates the model at a physically consistent equilibrium point,
- Generates the numerical state-space matrices,
- Verifies observability,
- Designs both the stabilizing LQR controller and the state observer,
- Builds the augmented controller with integral action.

The script therefore acts as the analytical bridge between the nonlinear Simulink plant and the linear control architecture implemented in the simulation model.

1. Symbolic Parameters

In the first section, all physical parameters of the half-car model are declared symbolically:

```
%% 1. SYMBOLIC PARAMETERS
syms m kf kr betaf betar ell0 g df dr J mwf mwr ktf ktr hcg gammaf gammar real
vars = [m kf kr betaf betar ell0 g df dr J mwf mwr ktf ktr hcg gammaf gammar];


```

This step ensures that the subsequent Jacobian computation remains fully analytical.

Keeping parameters symbolic allows:

- Exact differentiation of nonlinear expressions,
- Clear separation between model structure and numerical values,
- Reusability of the script for different vehicle configurations.

2. State Variables

The full nonlinear state vector is defined as:

```
%% 2. STATE VARIABLES (8x1 Column)
syms pz vz theta omega z_wf vz_wf z_wr vz_wr real
states = [pz; vz; theta; omega; z_wf; vz_wf; z_wr; vz_wr];

```

This 8-state formulation includes:

- Sprung mass vertical dynamics (p_z, v_z),
- Pitch dynamics (θ, ω),
- Front and rear unsprung mass dynamics.

Defining the states explicitly in symbolic form ensures consistency between:

- Nonlinear dynamics,
- Measurement equations,
- Performance outputs,
- Linearized state-space matrices.

3. Control Inputs

The control input vector is defined as:

```
%% 3. CONTROL INPUTS (2x1 Column)
syms u_F u_M real
inputs_u = [u_F; u_M];
```

These represent:

- u_F: total vertical control force,
- u_M: control pitching moment.

The symbolic formulation allows correct front/rear force redistribution according to the geometric distances d_f and d_r.

4. Disturbances

The disturbance vector includes:

```
%% 4. DISTURBANCES (6x1 Column)
syms zroad_f dzroad_f zroad_r dzroad_r Fxf Fxr real
inputs_w = [zroad_f; dzroad_f; zroad_r; dzroad_r; Fxf; Fxr];
```

This formulation is particularly important because it separates:

- Vertical road excitations,
- Longitudinal wheel forces.

Such separation enables independent simulation of:

- Road-induced vibrations,
- Acceleration and braking scenarios,
- Combined vertical and longitudinal disturbances.

5. Sensor Noise and References

Sensor noise variables n_i are introduced symbolically in the measurement equations.

Reference signals r_z and r_θ are defined separately to ensure that the linearization process is performed around equilibrium without embedding reference offsets directly into the Jacobians.

```
%% 5. SENSOR NOISE & REFS
syms n1 n2 n3 n4 n5 rz_ref rtheta_ref real
noise = [n1; n2; n3; n4; n5];
refs  = [rz_ref; rtheta_ref];

```

This separation is crucial for correct controller synthesis, since references are handled as exogenous inputs in Simulink rather than as part of the plant dynamics.

6. Nonlinear Kinematics and Dynamics

This section constructs the core nonlinear dynamics of the half-car model.

```
%% 6. KINEMATICS & DYNAMICS
s1 = (pz + df * sin(theta)) - z_wf;
s3 = (pz - dr * sin(theta)) - z_wr;
s2 = vz + df * omega * cos(theta) - vz_wf;
s4 = vz - dr * omega * cos(theta) - vz_wr;

fsf = -kf*s1 - betaf*s2;
fsr = -kr*s3 - betar*s4;
F_v_long_f = gammaf * Fxf;
F_v_long_r = gammar * Fxr;
F_tot_x    = Fxf + Fxr;

ftf = ktf*(zroad_f - z_wf);
ftr = ktr*(zroad_r - z_wr);

faf = (u_M + dr*u_F)/(df + dr);
far = (df*u_F - u_M)/(df + dr);

% Equazioni di stato (dxdt)
f2 = -g + (fsf + fsr + faf + far + F_v_long_f + F_v_long_r)/m;
f4 = (df*(fsf + faf + F_v_long_f) - dr*(fsr + far + F_v_long_r) + F_tot_x*hcg)/J;
f_wf = (ftf - fsf - faf - F_v_long_f - mwf*g)/mwf;
f_wr = (ftr - fsr - far - F_v_long_r - mwr*g)/mwr;

f_sys = [vz; f2; omega; f4; vz_wf; f_wf; vz_wr; f_wr];

```

Relative velocities s_2 and s_4 include the nonlinear coupling term $\omega \cos\theta$, ensuring correct pitch-to-heave interaction.

Spring-damper forces are then defined, followed by:

- Tire stiffness forces,
- Active force redistribution,

- Longitudinal load transfer contributions.

Finally, the full nonlinear state equation:

$$\dot{x} = f(x, u, w)$$

is assembled in vector form.

This formulation is identical to the one implemented in the Simulink nonlinear plant block, guaranteeing coherence between symbolic derivation and simulation model.

7. Output Definition: Sensors and Performance Channels

After defining the nonlinear dynamics, the script introduces two distinct output vectors:

- The measurement vector $y(x, u, w)$,
- The performance vector $z(x, u, w)$.

```
%% 7. OUTPUTS (Sensori y e Performance z)
ax_i = F_tot_x/m;
az_i = f2;
az_r = az_i + g;

y_sys = [sin(theta)*az_r + cos(theta)*ax_i + n1;
         cos(theta)*az_r - sin(theta)*ax_i + n2;
         omega + n3;
         s1 + n4;
         s3 + n5];

z_sys = [pz; theta; omega; s1; s3; vz];

```

Measurement Vector The sensor output is defined as:

$$y = \begin{bmatrix} a_y \\ a_z \\ \omega \\ s_1 \\ s_3 \end{bmatrix},$$

where the accelerometer signals are reconstructed from inertial accelerations:

$$a_x = \frac{F_{xf} + F_{xr}}{m}, \qquad a_z = f_2 + g.$$

These accelerations are then projected into the body reference frame using $\sin\theta$ and $\cos\theta$, reproducing the nonlinear coupling between pitch angle and measured vertical acceleration.

Sensor noise terms n_i are explicitly included in symbolic form. This is essential for the subsequent observer design, since the noise covariance matrix R_{obs} directly reflects this structure.

Performance Vector The performance vector is defined as:

$$z = \begin{bmatrix} p_z \\ \theta \\ \omega \\ s_1 \\ s_3 \\ v_z \end{bmatrix}.$$

Importantly, reference subtraction is not embedded in this symbolic definition. References enter the Simulink model externally as offsets, ensuring that the Jacobian matrices are computed with respect to the true physical states rather than tracking errors.

This separation preserves correctness of the linearization process.

8. Jacobian Computation

The core purpose of the script is the computation of all required Jacobian matrices.

```
%% 8. JACOBIANS (Calcolo di tutte le matrici)
A_sym   = jacobian(f_sys, states);
B1_sym  = jacobian(f_sys, inputs_u);
B2_sym  = jacobian(f_sys, inputs_w);

C_sym   = jacobian(y_sys, states);
D1_sym  = jacobian(y_sys, inputs_u);
D2_sym  = jacobian(y_sys, inputs_w);

CE_sym  = jacobian(z_sys, states);
DE1_sym = jacobian(z_sys, inputs_u);
DE2_sym = jacobian(z_sys, inputs_w);

```

The following symbolic derivatives are computed:

$$A = \frac{\partial f}{\partial x}, \quad B_1 = \frac{\partial f}{\partial u}, \quad B_2 = \frac{\partial f}{\partial w},$$

$$C = \frac{\partial y}{\partial x}, \quad D_1 = \frac{\partial y}{\partial u}, \quad D_2 = \frac{\partial y}{\partial w},$$

$$C_E = \frac{\partial z}{\partial x}, \quad D_{E1} = \frac{\partial z}{\partial u}, \quad D_{E2} = \frac{\partial z}{\partial w}.$$

This step guarantees that the linearized model is obtained through exact analytical differentiation, avoiding numerical approximation errors.

The resulting structure is fully consistent with the generalized plant formulation:

$$\dot{x} = Ax + B_1 u + B_2 w,$$
$$y = Cx + D_1 u + D_2 w,$$
$$z = C_E x + D_{E1} u + D_{E2} w.$$

9. Numerical Evaluation at the Equilibrium Point

After symbolic derivation, all expressions are numerically evaluated at a physically consistent static equilibrium.

```
%% 9. NUMERICAL EVALUATION
m_n = 2550; kf_n = 35000; kr_n = 35000; betaf_n = 4200; betar_n = 4200;
ell0_n = 0.55; g_n = 9.81; df_n = 1.65; dr_n = 1.65; J_n = m_n*(0.38*(df_n+dr_n))^2;
mwf_n = 48; mwr_n = 48; ktf_n = 270000; ktr_n = 270000; hcg_n = 0.60;
gammaf_n = 0.00; gammar_n = 0.00;

p_num = [m_n, kf_n, kr_n, betaf_n, betar_n, ell0_n, g_n, df_n, dr_n, J_n, mwf_n,
    mwr_n, ktf_n, ktr_n, hcg_n, gammaf_n, gammar_n];

% Calcolo Punto di Equilibrio (Vettore COLONNA)
pz_static = -(m_n * g_n) / (kf_n + kr_n);
h_static  = -((m_n + mwf_n + mwr_n) * g_n) / (2 * ktf_n);

pz0 = pz_static + h_static;
th0 = 0;

x0 = [pz_static + h_static; 0; 0; 0; h_static; 0; h_static; 0];

x0_lin_en = 0;


% Sostituzione numerica
sub_map = [states; inputs_u; inputs_w; noise; refs];
sub_val = [x0; 0;0; 0;0;0;0;0;0; 0;0;0;0;0; 0;0];

clean_sub = @(S) double(vpa(subs(subs(S, vars, p_num), sub_map, sub_val), 10));
```

The equilibrium configuration includes:

- Gravitational preload of the suspension,
- Static tire deflection,
- Zero pitch angle,
- Zero velocities.

The vertical equilibrium position is computed and tire compression is determined from total static load distribution.

11. Numerical matrices

Substituting both parameter values and equilibrium state into the symbolic Jacobians produces the numerical matrices:

These matrices are exported to the MATLAB workspace and used directly in the Simulink linear control block.

```
%% 10. MATRICI FINALI
A_N   = clean_sub(A_sym);
B1_N  = clean_sub(B1_sym);
B2_N  = clean_sub(B2_sym);
C_N   = clean_sub(C_sym);
D1_N  = clean_sub(D1_sym);
D2_N  = clean_sub(D2_sym);
CE_N  = clean_sub(CE_sym);
DE1_N = clean_sub(DE1_sym);
DE2_N = clean_sub(DE2_sym);

% Calcolo dei valori di equilibrio per i sensori e le performance
y_eq = clean_sub(y_sys);
z_eq = clean_sub(z_sys);

disp('Linearizzazione completa eseguita con successo.');
```

11. Observability Analysis

Before designing the observer, the script verifies structural observability.

```
%% 11. VERIFICA OSSERVABILITÀ
% Calcolo della matrice di osservabilità per il sistema (A_N, C_N)
Ob = obsv(A_N, C_N);

% Verifica del rango
n_stati = size(A_N, 1);
rango_Ob = rank(Ob);

fprintf('--- Analisi Osservabilità ---\n');
fprintf('Numero di stati: %d\n', n_stati);
fprintf('Rango della matrice di osservabilità: %d\n', rango_Ob);

if rango_Ob == n_stati
    disp('RISULTATO: Il sistema è COMPLETAMENTE OSSERVABILE. Possiamo progettare l''Observer.');
else
    disp('ATTENZIONE: Il sistema NON è completamente osservabile. L''Observer non potrà stimare tutti gli stati.');
    % Analisi dei sottospazi non osservabili (opzionale)
    [~, ~, ~, ex] = obsvf(A_N, B1_N, C_N);
end
```

The observability matrix is computed as:

$$\mathcal{O} = \begin{bmatrix} C_N \\ C_N A_N \\ C_N A_N^2 \\ \vdots \end{bmatrix}.$$

Its rank is compared with the number of states.

Full rank confirms that the selected sensor configuration (accelerometers, gyroscope, suspension deflections) allows reconstruction of all eight states.

This analytical verification ensures that the observer design is theoretically admissible.

12. LQR State Feedback Design

A stabilizing LQR controller is first designed using:

$$K_s = \text{lqr}(A_N, B_{1N}, Q_{lqr}, R_{lqr}).$$

The weighting matrix Q_{lqr} assigns:

- High weight to vertical displacement p_z,
- Even higher weight to pitch angle θ,
- Moderate weight to velocities v_z and ω,
- Low weight to unsprung mass dynamics.

The matrix R_{lqr} limits control aggressiveness and prevents unrealistic actuator demand.

This step provides a baseline stabilizing controller.

```
%% 12. TUNING STABILIZZATORE (LQR)
% Torniamo a valori più umani e bilanciamo meglio

q_pz    = 1e6;
q_theta = 5e6;    % Teniamo il pitch 5 volte più pesante del bounce
q_vz    = 1e2;    % Smorzamento (fondamentale per non farlo esplodere)
q_omega = 1e2;    % Smorzamento rotazionale

% Ricostruiamo Q
Q_lqr = diag([q_pz, q_vz, q_theta, q_omega, 10, 10, 10, 10]);

% Alziamo R: non sotto 1e-3.
% R funge da "freno" per evitare che il controllo diventi troppo nervoso.
r_u = 1e-3;
R_lqr = diag([r_u, r_u]);

[Ks, ~, ~] = lqr(A_N, B1_N, Q_lqr, R_lqr);


```

13. Observer Design via Dual LQR

The state observer gain K_o is computed using the dual LQR formulation:

$$K_o = \mathrm{lqr}(A_N^T, C_N^T, Q_{obs}, R_{obs})^T.$$

The tuning philosophy reflects sensor trust hierarchy:

- Accelerometers are heavily downweighted during longitudinal acceleration,
- The gyroscope is highly trusted for rotational dynamics,
- Suspension deflection sensors prevent long-term drift.

This produces a full-order observer capable of robust state estimation even during acceleration or braking maneuvers.

```
%% 13. PROGETTO OSSERVATORE (Fine Tuning Anti-Longitudinal)
% Q_obs: Incertezza del modello. Teniamo alto il valore sulla velocità
% del pitch (stato 4) per non perdere la reattività
Q_obs = diag([0.01, 100, 1, 1000, 10, 10, 10, 10]);

r_acc_fake = 1e4;    % Gli accelerometri mentono durante l'accelerazione ax
r_gyro_true = 1e-6;  % Il giroscopio è la verità sulla rotazione
r_lvdt_true = 5e-2;  % Le LVDT servono per non far driftare il segnale a fine spinta

R_obs = diag([r_acc_fake, r_acc_fake, r_gyro_true, r_lvdt_true, r_lvdt_true]);

% Calcolo del guadagno Ko tramite la dualità LQR (Metodo Di Ruscio/Balchen)

Ko = lqr(A_N', C_N', Q_obs, R_obs)';

```

14. Augmented LQR with Integral Action

To eliminate steady-state tracking error, integral action is introduced on:

- Heave error,
- Pitch error.

An augmented state vector is constructed:

$$x_{aug} = \begin{bmatrix} x \\ \int e_{heave} \\ \int e_{pitch} \end{bmatrix}.$$

The augmented system matrices are built as:

$$A_{aug} = \begin{bmatrix} A_N & 0 \\ C_{integ} & 0 \end{bmatrix}, \qquad B_{aug} = \begin{bmatrix} B_{1N} \\ 0 \end{bmatrix}.$$

The resulting gain matrix:

$$K_{total} = \begin{bmatrix} K_s & K_i \end{bmatrix}$$

is obtained through LQR applied to the augmented system.

Only the first two performance channels (heave and pitch) are integrated, ensuring stability while preserving damping of higher-order dynamics.

```
%% 14. PROGETTO CONTROLLO LQR AUMENTATO (Ks 2x8 + Ki 2x6) - STABILE
% --- 1. Parametri di Tuning ---
weight_P = 180;
weight_D = 60;
weight_I = 80;
% --- 2. Selezione Matrici
A_ctrl = A_N;
B_ctrl = B1_N;
% Selezioniamo solo Heave e Pitch per l'integrazione (Prime 2 righe di CE_N)
C_integ = CE_N(1:2, :);
n_da_integrare = 2;

% --- 3. Costruzione Sistema Aumentato (10 stati: 8 fisici + 2 integrali) --
A_aug = [ A_ctrl,           zeros(8, n_da_integrare);
          C_integ,          zeros(n_da_integrare, n_da_integrare) ];

B_aug = [ B_ctrl;
          zeros(n_da_integrare, 2) ];

% --- 4. R_aug (Sforzo di controllo) ---
% Alziamo leggermente R a 0.005 per stabilizzare il calcolo Riccati
R_aug = eye(2) * 0.005;
```

```
% --- 5. Definizione dei Pesi Base ---
q_pz_base      = 1e7;
q_ptheta_base  = 1e7;
q_vz_base      = 5e7;
q_vtheta_base  = 8e7;
q_unsprung     = 1e0;
q_heave_int_base = 1e9;
q_pitch_int_base = 1e9;

% --- 6. Composizione Matrice Q_aug (10x10) ---
Q_diag = [q_pz_base * weight_P, ...     % pz
          q_vz_base * weight_D, ...     % vz
          q_ptheta_base * weight_P, ... % theta
          q_vtheta_base * weight_D, ... % omega
          q_unsprung, q_unsprung, ...   % ruote pos
          q_unsprung, q_unsprung, ...   % ruote vel
          q_heave_int_base * weight_I, ... % Integrale Heave
          q_pitch_int_base * weight_I];   % Integrale Pitch

Q_aug = diag(Q_diag);

% --- 7. Calcolo Guadagno LQR ---
K_total = lqr(A_aug, B_aug, Q_aug, R_aug);
```

```
% Ks (2x8): Guadagno sugli stati fisici
Ks = K_total(:, 1:8);

% Ki_small (2x2): Guadagno sugli integrali calcolati
Ki_small = K_total(:, 9:10);

% Ki (2x6): Espansione per il tuo blocco Simulink che legge 6 errori
% Mettiamo i guadagni reali sulle prime due colonne e zero sulle altre 4.
Ki = zeros(2, 6);
Ki(:, 1:2) = Ki_small;
```

15. Final Outcome of the Script

At the end of the execution, the script provides:

- Linearized state-space matrices,
- Observer gain K_o,
- State feedback gain K_s,
- Integral gain K_i.

These quantities are directly used inside the Simulink linear control block, ensuring full consistency between:

- Analytical derivation,
- Controller synthesis,
- Time-domain simulation.

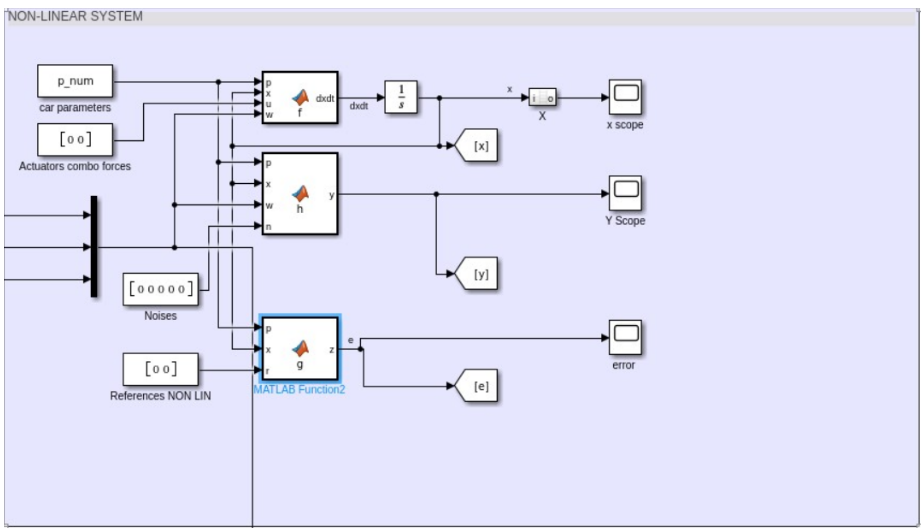
The script therefore constitutes the mathematical backbone of the entire control implementation.

2.0.3 Nonlinear Half-Car Model

The nonlinear plant implemented in Simulink represents the high-fidelity reference model used for validation of the control architecture. In order to preserve modularity and analytical clarity, the plant is structured into three distinct MATLAB Function blocks:

- $f(p, x, u, w)$: nonlinear state dynamics,
- $h(p, x, w, n)$: sensor model,
- $g(p, x, r)$: performance output vector.

This separation mirrors the theoretical formulation and ensures a clean distinction between physical dynamics, measurable outputs, and control performance variables.



State Dynamics Block $f(p, x, u, w)$

The block f implements the complete nonlinear 8-state half-car model:

$$x = \begin{bmatrix} p_z & v_z & \theta & \omega & z_{wf} & v_{zf} & z_{wr} & v_{zr} \end{bmatrix}^T,$$

including sprung and unsprung mass dynamics.

Nonlinear Kinematics Suspension deflections are computed using exact nonlinear geometry:

$$s_1 = (p_z + d_f \sin\theta) - z_{wf}, \qquad s_3 = (p_z - d_r \sin\theta) - z_{wr},$$

while relative velocities include pitch–heave coupling:

$$s_2 = v_z + d_f \omega \cos\theta - v_{zf}, \qquad s_4 = v_z - d_r \omega \cos\theta - v_{zr}.$$

The use of $\sin\theta$ and $\cos\theta$ guarantees correct large-angle behavior and preserves geometric consistency.

```
%% 4. KINEMATICS
s1 = (pz + df * sin(ptheta)) - hwheelf;
s3 = (pz - dr * sin(ptheta)) - hwheelr;
s2 = vz + df * vtheta * cos(ptheta) - dhwheelf;
s4 = vz - dr * vtheta * cos(ptheta) - dhwheelr;

```

Suspension and Tire Forces Spring–damper forces are modeled as:

$$f_{sf} = -k_f s_1 - \beta_f s_2, \qquad f_{sr} = -k_r s_3 - \beta_r s_4.$$

Tire forces are represented through linear stiffness elements:

$$f_{tf} = k_{tf}(z_{road,f} - z_{wf}), \qquad f_{tr} = k_{tr}(z_{road,r} - z_{wr}).$$

```
%% 5. DYNAMICS & FORCES
fsf = -kf*s1 - betaf*s2;
fsr = -kr*s3 - betar*s4;

F_v_long_f = gammaf * Fxf;
F_v_long_r = gammar * Fxr;

ftf = ktf*(zroad_f - hwheelf);
ftr = ktr*(zroad_r - hwheelr);

% Distribuzione attuatori (u(1)=F_total, u(2)=M_total)
faf = (u(2) + dr*u(1))/(df + dr);
far = (df*u(1) - u(2))/(df + dr);

```

Actuator Force Distribution The control inputs are defined as total vertical force u_F and total pitching moment u_M. These are redistributed between front and rear according to vehicle geometry:

$$f_{af} = \frac{u_M + d_r u_F}{d_f + d_r}, \qquad f_{ar} = \frac{d_f u_F - u_M}{d_f + d_r}.$$

This guarantees that the pair (u_F, u_M) produces the desired resultant force and moment about the center of gravity.

Longitudinal Coupling and Load Transfer Longitudinal wheel forces F_{xf} and F_{xr} influence vertical and rotational dynamics through:

- Geometric coupling coefficients γ_f, γ_r (anti-dive / anti-squat),
- The pitching moment term $F_{tot,x}h_{cg}$.

This introduces realistic coupling between acceleration/braking maneuvers and vertical ride dynamics.

State Equations The accelerations are computed as:

$$\dot{v}_z = -g + \frac{f_{sf} + f_{sr} + f_{af} + f_{ar} + F_{v,long,f} + F_{v,long,r}}{m},$$

$$\dot{\omega} = \frac{d_f(f_{sf} + f_{af} + F_{v,long,f}) - d_r(f_{sr} + f_{ar} + F_{v,long,r}) + F_{tot,x}h_{cg}}{J}.$$

Unsprung mass accelerations are derived from tire–suspension interaction.

The block finally returns:

$$\dot{x} = f(p, x, u, w),$$

which is numerically integrated within Simulink.

This block constitutes the nonlinear ground-truth model used in all simulations.

Measurement Block $h(p, x, w, n)$

The block h models the sensor suite installed on the vehicle, including:

- Two accelerometer channels,
- One gyroscope,
- Two suspension deflection sensors (LVDT-type).

Inertial Accelerations The inertial accelerations of the center of gravity are first computed in the inertial frame:

```
%% 5. ACCELERAZIONI INERZIALI (Riferimento Terrestre)
% Accelerazione verticale "pura" del CG (senza gravità)
az_inertial = -g + (fsf + fsr + F_v_long_f + F_v_long_r)/m;

% Accelerazione longitudinale del CG dovuta alle forze alle ruote
ax_inertial = Fx_tot / m;

az_rel = az_inertial + g;

```

Since accelerometers measure reaction forces rather than pure inertial acceleration, gravity is reintroduced:

$$a_z^{rel} = a_z^{inertial} + g.$$

Projection to Body Frame The IMU is rigidly attached to the chassis and therefore rotates with θ. The inertial accelerations are projected into the body frame:

```
%% 6. PROIEZIONE NEL SISTEMA DI RIFERIMENTO CORPO (Body Frame)

ax_body =  sin(theta)*az_rel + cos(theta)*ax_inertial;
az_body =  cos(theta)*az_rel - sin(theta)*ax_inertial;
```

This formulation ensures consistency with the apparent pitch angle reconstruction previously introduced.

Sensor Output Vector The complete measurement vector is:

$$y = \begin{bmatrix} a_x^{body} + n_1 \\ a_z^{body} + n_2 \\ \omega + n_3 \\ s_1 + n_4 \\ s_3 + n_5 \end{bmatrix}.$$

Noise terms n_i are explicitly included to enable realistic observer testing.

Performance Block $g(p, x, r)$

The block g defines the performance vector used by the controller.

References are provided externally:

$$r = \begin{bmatrix} r_z \\ r_\theta \end{bmatrix}.$$

Suspension deflections are recomputed kinematically:

$$s_f = (p_z + d_f \sin\theta) - z_{wf}, \qquad s_r = (p_z - d_r \sin\theta) - z_{wr}.$$

The resulting performance vector is:

$$z = \begin{bmatrix} p_z - r_z \\ \theta - r_\theta \\ \omega \\ s_f \\ s_r \\ v_z \end{bmatrix}.$$

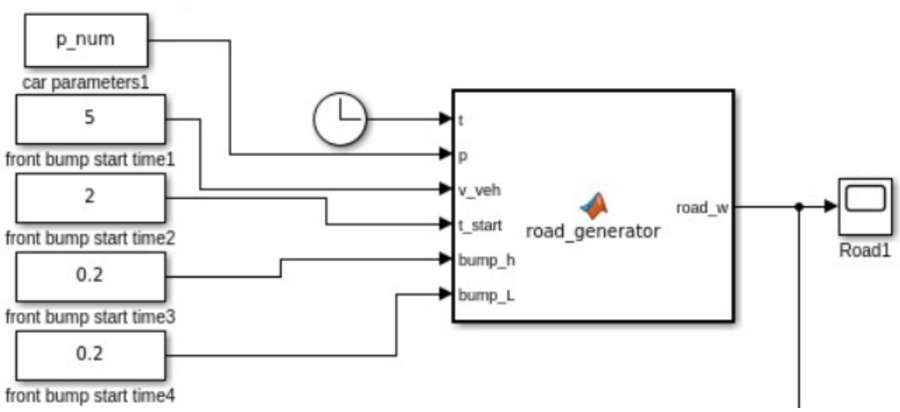
The first two components enforce reference tracking (heave and pitch), while the remaining components:

- Penalize rotational oscillations (ω),
- Monitor suspension stroke limits (s_f, s_r),

- Provide heave damping through v_z.

This block therefore bridges the nonlinear plant and the control law, defining exactly which physical quantities are regulated or minimized.

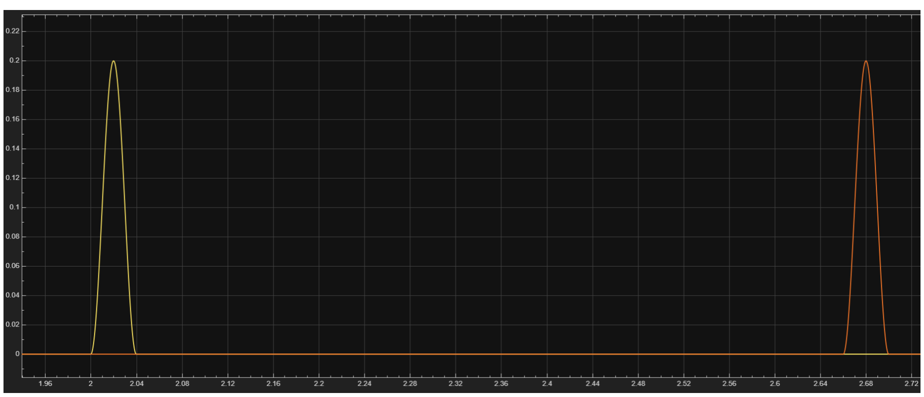
2.0.4 Road Profile Generator



The road disturbance subsystem generates the front and rear road profiles:

$$z_{road,f}(t), \quad z_{road,r}(t),$$

and their derivatives.



The block is fully parametric and allows independent control of:

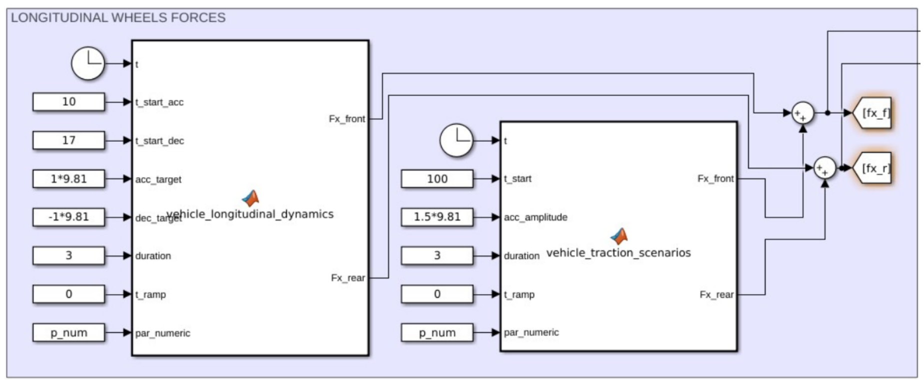
- Bump height,
- Bump longitudinal length,
- Vehicle approach speed,

- Simulation start time of the disturbance.

This enables reproducible and configurable excitation scenarios, including single bumps, repeated profiles, or asymmetric front-rear excitation.

The rear profile is automatically delayed according to the wheelbase and vehicle speed to maintain physical consistency.

2.0.5 Longitudinal Wheel Forces Block



To simulate acceleration and braking maneuvers, a dedicated block generates:

$$F_{xf}, \quad F_{xr}.$$

This subsystem allows:

- Independent amplitude selection,
- Custom temporal profiles (step, ramp, shaped inputs),
- Adjustable front/rear distribution,
- Simulation of traction or braking scenarios.

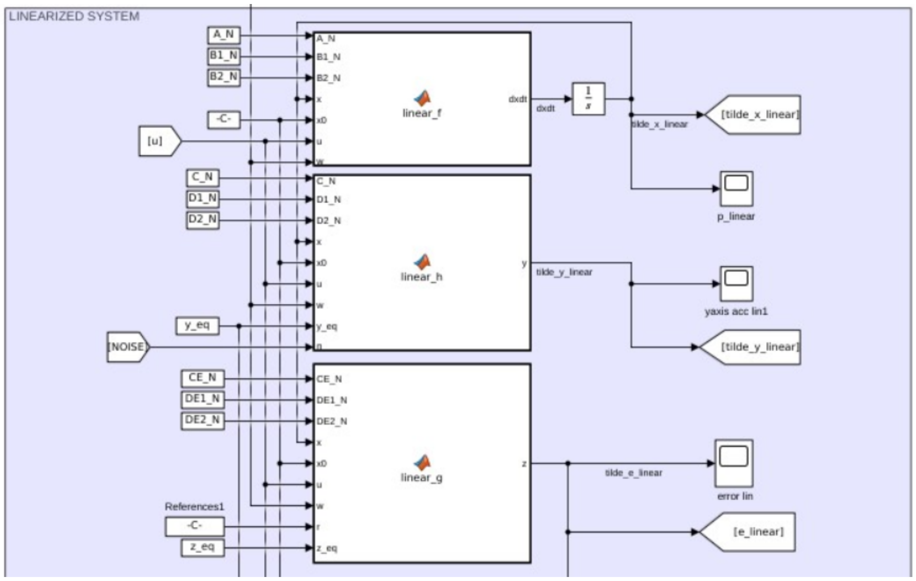
Longitudinal forces influence both vertical dynamics (through load transfer) and pitch dynamics via the h_{cg} coupling term in the rotational equation.

This block is essential to evaluate controller robustness during combined vertical and longitudinal excitations.

2.0.6 Linearized Model

The linear model is obtained via symbolic Jacobian computation around the static equilibrium point.

The linear state-space representation is:

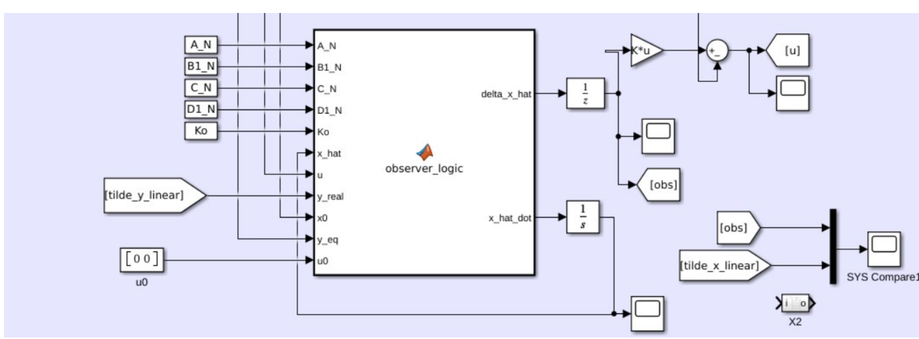


$$\dot{x} = A_N x + B_{1N} u + B_{2N} w,$$
$$y = C_N x + D_{1N} u + D_{2N} w,$$
$$z = C_{E,N} x + D_{E1,N} u + D_{E2,N} w.$$

The equilibrium is computed including gravitational preload and tire deflection, ensuring accurate linearization around realistic operating conditions.

The observability matrix has full rank, confirming that all states are reconstructible from the selected sensor configuration.

2.0.7 State Observer



A full-order observer is implemented using the dual LQR formulation:

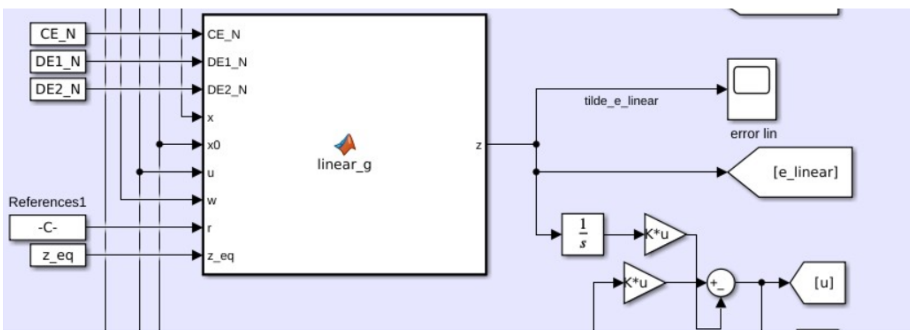
$$K_o = \text{lqr}(A_N^T, C_N^T, Q_{obs}, R_{obs})^T.$$

The observer tuning intentionally:

- Downweights accelerometers during longitudinal acceleration,
- Assigns high confidence to the gyroscope,
- Uses LVDT suspension measurements to prevent drift.

The observer provides $\hat{x}$ for feedback in the linear controller.

2.0.8 Augmented LQR Controller



An augmented state vector including integral action on heave and pitch errors is constructed:

$$x_{aug} = \begin{bmatrix} x \\ \int e_{heave} \\ \int e_{pitch} \end{bmatrix}.$$

The resulting control law is:

$$u = -K_s x - K_i e,$$

where:

- $K_s \in \mathbb{R}^{2\times 8}$,
- $K_i \in \mathbb{R}^{2\times 6}$ (with only the first two channels active).

The LQR weights are tuned to balance:

- Heave regulation,
- Pitch stabilization,
- Vertical and rotational damping,
- Limited control effort.

2.0.9 Simulation Management and Visualization

Enable switches are used to:

- Activate/deactivate longitudinal forces,
- Enable road disturbances,
- Switch between nonlinear and linear models,
- Compare open-loop and closed-loop responses.

Dedicated scopes monitor:

- Vertical displacement and pitch,
- Suspension deflections,
- Control forces,
- Estimated versus true states.

This modular implementation allows systematic validation of performance, robustness, and observer accuracy under different operating conditions.

2.1 Vehicle Parameters and Numerical Setup

To evaluate the performance of the proposed active suspension control, a numerical setup representative of a luxury high-end sedan (e.g., a Rolls-Royce Ghost or similar) was implemented. The choice of parameters aims to replicate a "Magic Carpet" ride quality, characterized by low natural frequencies and high isolation from road disturbances.

2.1.1 Mass and Geometry

The vehicle body mass is set to $m = 2550$ kg, reflecting the significant weight of luxury vehicles. A perfectly balanced weight distribution is assumed, with $d_f = d_r = 1.65$ m, resulting in a total wheelbase of $L = 3.3$ m.

The pitch moment of inertia J is not assumed as a point mass but calculated using the radius of gyration k_p. For luxury sedans, k_p typically ranges between 35% and 40% of the wheelbase. In this model, we used:

$$J = m \cdot (0.38 \cdot L)^2 \approx 4016 \text{ kg} \cdot \text{m}^2 \tag{2.1}$$

This high inertia value contributes to a more stable and "slower" pitch response, which is perceived as more comfortable by passengers.

2.1.2 Suspension and Tire Characteristics

The stiffness and damping coefficients are tuned to be relatively soft:

- **Spring Stiffness (k_f, k_r):** Values of $35,000$ N/m are chosen to maintain a low natural frequency (around 1 Hz), which is the human comfort benchmark.
- **Damping (β_f, β_r):** Set at 4200 N·s/m. In a real active system, this would represent the "base" passive damping upon which the actuators exert additional forces.
- **Unsprung Masses (m_{wf}, m_{wr}):** Fixed at 48 kg per wheel, representing large, high-profile luxury wheels.
- **Tire Stiffness (k_{tf}, k_{tr}):** The value of $270,000$ N/m represents high-profile tires which provide additional filtering of high-frequency road noise.

2.1.3 Anti-Pitch Geometry

The anti-geometry coefficients are chosen to provide a subtle but effective correction without compromising the suspension's ability to absorb bumps:

- **Anti-Dive ($\gamma_f = 0.05$):** A 5% anti-dive geometry reduces front-end dip during braking.
- **Anti-Squat ($\gamma_r = 0.08$):** An 8% anti-squat geometry prevents excessive rear-end compression during acceleration.

Table 2.1: Numerical Parameters for the Luxury Sedan Model.

Symbol	**Description**	**Value**	**Unit**
m	Sprung Mass	2550	kg
J	Pitch Inertia	4016	kg·m^2
d_f, d_r	CG to Axle Distance	1.65	m
k_f, k_r	Suspension Stiffness	35000	N/m
β_f, β_r	Passive Damping	4200	N·s/m
m_{wf}, m_{wr}	Unsprung Mass	48	kg
k_{tf}, k_{tr}	Tire Stiffness	270000	N/m
h_{cg}	CG Height	0.60	m
γ_f, γ_r	Anti-Dive/Squat Coeff.	0.05, 0.08	-

Chapter 3

Control Design

3.1 Proposed Control Architecture

The control strategy adopted for the Rolls-Royce half-car model is based on a **Multi-Loop Output Feedback** architecture, combining optimal state regulation with stochastic estimation. The overall block diagram of the proposed solution is illustrated in Figure

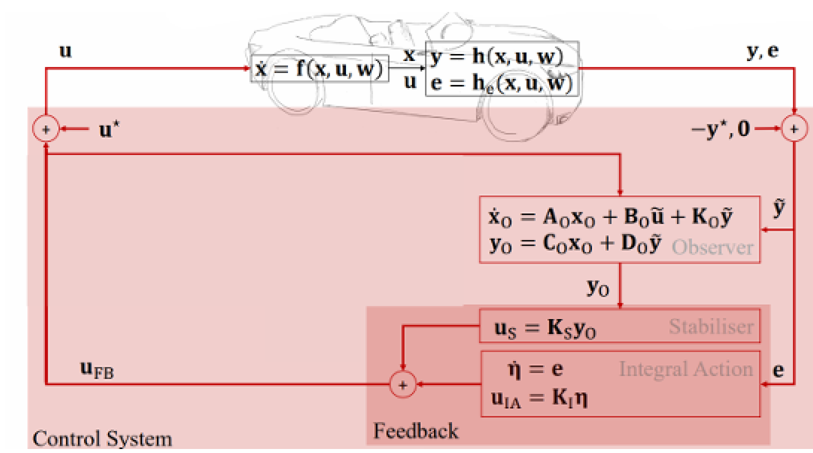


Figure 3.1: Control Architecture

The scheme consists of three main functional blocks:

1. **The Augmented Plant:** Represents the physical vehicle dynamics coupled with the mathematical models of the road disturbances (integrated as stochastic states).
2. **The Kalman Observer:** A state estimator that reconstructs the full state vector $\hat{x}$ (including unmeasurable velocities and road profiles) from

noisy sensor measurements y (suspension deflections and accelerations).

3. **The LQR-Integrator Regulator:** A feedback controller that combines the estimated state $\hat{x}$ for stabilization and damping, and the integral of the error η for set-point tracking.

3.1.1 Design Choice: Implicit Disturbance Rejection vs. Explicit Feed-Forward

A critical design choice in this project was the exclusion of an explicit *Feed-Forward* branch based on "preview" sensors (such as LiDAR or cameras scanning the road ahead). While Preview Control is a common technique in modern active suspensions to prepare the vehicle for upcoming bumps, it was discarded in this architecture for two main reasons:

1. **Hardware Complexity and Cost:** Explicit feed-forward requires additional expensive environmental sensors and high computational power to process road surface data in real-time.

2. **Disturbance Observation Capability:** The designed Kalman Filter is not limited to filtering noise but is configured as a *Disturbance Observer*. By including the road profile states ($z_{road}, \dot{z}_{road}$) in the system model and tuning the process noise covariance matrix (Q_{kal}) aggressively, the observer is capable of estimating the road input virtually instantaneously.

Consequently, the control law $u = -K\hat{x}$ acts on the estimated road states $\hat{\epsilon}$ immediately as the wheel encounters the irregularity. This mechanism creates an **implicit feed-forward action**: the controller "feels" the bump through the unsprung mass dynamics and applies a counter-force before the disturbance propagates significantly to the chassis, achieving the "Magic Carpet" performance without the need for look-ahead sensors.

Le specifiche di progetto includono:

- **Inseguimento del set-point:** Mantenimento dell'altezza statica e dell'assetto orizzontale ($\theta = 0$) anche in presenza di carichi costanti.
- **Reiezione dei disturbi:** Eliminazione dell'errore a regime indotto da irregolarità del manto stradale o forze longitudinali.
- **Dinamica Anti-Pitch:** Compensazione attiva degli effetti di *Anti-Dive* e *Anti-Squat* introdotti dalla geometria delle sospensioni (parametri γ_f, γ_r).

3.2 Architettura di Controllo LQR-PID Aumentata

To ensure the elimination of steady-state error for the controlled variables (heave p_z and pitch θ), the linearized model was extended with two integrator states.

3.2.1 Augmented System and LQR Synthesis

Let $x_{ctrl} \in \mathbb{R}^8$ be the vector of controllable physical states (excluding road kinematics). We define the error vector $e = [e_z, e_\theta]^T$ as the deviation from the reference heights. We introduce the integral state vector $\eta(t) = \int_0^t e(\tau)d\tau$. The augmented system takes the form:

$$\begin{bmatrix} \dot{x}_{ctrl} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} A_{ctrl} & 0_{8\times2} \\ C_{E,ctrl} & 0_{2\times2} \end{bmatrix} \begin{bmatrix} x_{ctrl} \\ \eta \end{bmatrix} + \begin{bmatrix} B_{ctrl} \\ D_{E1,ctrl} \end{bmatrix} u \tag{3.1}$$

The control gain $K_{aug} = [K_{prop} \quad K_{int}]$ is obtained by minimizing the quadratic cost functional:

$$J = \int_0^\infty (x_{aug}^T Q_{aug} x_{aug} + u^T R_{aug} u)dt \tag{3.2}$$

The weight matrix Q_{aug} was calibrated with a diagonal structure to weigh the proportional, derivative, and integral actions separately:

- **Heave Control:** $q_{pz} = 2 \cdot 10^8$ (P) and $q_{vz} = 8 \cdot 10^5$ (D). The integral action is heavily weighted with $q_{\int z} = 10^{10}$ to force leveling.
- **Pitch Control:** $q_{p\theta} = 3 \cdot 10^9$ (P) and $q_{v\theta} = 10^6$ (D). The integral weight $q_{\int \theta} = 5 \cdot 10^{11}$ is the highest to counteract rotational inertia J and the effects of pitching during braking/acceleration.
- **Unsprung Masses:** $q_{unsprung} = 10^2$, a deliberately low weight to allow the wheels to follow the road profile without transmitting excessive forces to the body.

3.3 State Estimation: Optimal Kalman Filter Theory

Full implementation of the LQR control law requires knowledge of the entire state vector $x \in \mathbb{R}^{12}$. However, in the real system, only a subset of these variables can be measured using sensors (accelerometers and potentiometers). To overcome this, a Kalman filter (KF) has been designed, which acts as an optimal observer in the presence of stochastic noise.

3.3.1 Stochastic Formulation and ϵ Disturbances

The linearized dynamic system is extended to include model and road surface uncertainties, collectively defined as ϵ. The state space model takes the form:

$$\begin{cases} \dot{x} = A_N x + B_1 u + Gw \\ y = C_N x + D_1 u + v \end{cases} \tag{3.3}$$

where $w \sim \mathcal{N}(0, Q_{kal})$ represents the process noise (which includes the stochastic variations of the road profile z_{road}) and $v \sim \mathcal{N}(0, R_{kal})$ represents the white noise affecting the sensors.

The states $x_9, \ldots, x_{12}$ (i.e., θ_{road} and z_{road} for the front and rear axles) are not subject to direct control inputs but evolve according to the characteristics of the terrain. The Kalman Filter estimates these "invisible" parameters by analyzing the prediction error on the measurable outputs y.

3.3.2 Observer Dynamics and Innovation

The observer reconstructs the estimated state $\hat{x}$ by minimizing the covariance of the estimation error $P = E[(x - \hat{x})(x - \hat{x})^T]$. The structure of the observer is:

$$\dot{\hat{x}} = A_N \hat{x} + B_1 u + K_o \underbrace{(y - \hat{y})}_{\text{Innovation}} \tag{3.4}$$

The **innovation** term $(y - C_N \hat{x} - D_1 u)$ is the difference between the actual measurement provided by the sensors and the measurement predicted by the model. The gain K_o (calculated in the code using the `lqr` command exploiting duality) determines the weight of the innovation:

- If $Q_{kal} \gg R_{kal}$ (as in the 10^6 vs 10^{-3} setup), the observer considers the model less reliable than the sensors. Consequently, the gain K_o will be high, making the estimation of ϵ extremely fast in following road bumps.
- If $R_{kal} \gg Q_{kal}$, the observer would filter the signals more, but at the risk of "losing" the impulsive component of road disturbances.

3.3.3 Road Profile Reconstruction (Disturbance Observation)

The main advantage of this approach is that the control force u is also calculated as a function of $\hat{\epsilon}$. When a wheel encounters an obstacle, the sudden acceleration of the unsprung mass generates a high innovation in the filter. The latter instantly corrects the estimate of $\hat{z}_{road,f}$ or $\hat{z}_{road,r}$.

The LQR controller, seeing this change in the estimated state, commands the actuators to apply a counterforce even before the disturbance propagates to the main body (sprung mass), effectively performing a sort of **feed-forward (implicit feed-forward)** based on the optimal estimate.

3.3.4 Stability of the Observer

The stability of the estimation is guaranteed by the fact that the pair (A_N, C_N) is observable. In the code, regularization has been applied to the matrix A (`A_reg = A_N - 1e-6*eye(12)`) to prevent numerical instability of the filter during periods of stationarity, ensuring that the eigenvalues of the estimation error $\text{eig}(A_N - K_o C_N)$ have a sufficiently large negative real part to guarantee rapid convergence.

3.4 modeling of Anti-Dive and Anti-Squat Geometry

In a standard half-car model, the suspension struts are often assumed to be perfectly vertical. However, in real vehicle suspension kinematics (e.g., double wishbone or multi-link setups), the instant centers of rotation are designed to create a geometric coupling between longitudinal forces and vertical body forces. This design feature is known as *Anti-Dive* (for the front axle during braking) and *Anti-Squat* (for the rear axle during acceleration).

3.4.1 Geometric Forces Definition

The anti-geometry effects are modeled by introducing two dimensionless coefficients, γ_f and γ_r (represented in the MATLAB code as `gammaf` and `gammar`). These coefficients determine the percentage of the longitudinal force that is converted into a vertical force acting on the sprung mass, bypassing the springs and dampers.

Let $F_{x,f}$ and $F_{x,r}$ be the longitudinal forces at the front and rear tire contact patches respectively (variables `fwfront` and `fwrear`). The vertical forces induced by the suspension geometry, denoted as $F_{v,f}^{geo}$ and $F_{v,r}^{geo}$, are defined as:

$$F_{v,f}^{geo} = \gamma_f \cdot F_{x,f} \tag{3.5}$$

$$F_{v,r}^{geo} = \gamma_r \cdot F_{x,r} \tag{3.6}$$

These forces provide a "stiffening" effect against pitch motion without actually increasing the spring stiffness k_f or k_r, thus preserving ride comfort during steady-state driving while reducing body rotation during aggressive longitudinal maneuvers.

3.4.2 Modified Equations of Motion

The introduction of these geometric forces and the consideration of the center of gravity height (h_{cg}) significantly modify the equilibrium and the dynamics of the sprung mass.

Vertical Dynamics (Heave)

The equation for the vertical acceleration $\ddot{z}$ (variable `f2`) must now account for the geometric lift forces. The Newton's second law for the vertical direction becomes:

$$m\ddot{z} = F_{sf} + F_{sr} + F_{af} + F_{ar} + F_{v,f}^{geo} + F_{v,r}^{geo} - mg \tag{3.7}$$

Where:

- F_{sf}, F_{sr} are the passive suspension forces (spring + damper).

- F_{af}, F_{ar} are the active control forces.
- $F_{v,f}^{geo}, F_{v,r}^{geo}$ are the anti-dive/anti-squat contributions.

Rotational Dynamics (Pitch)

The pitch dynamics are the most affected by these changes. The moment balance equation for $\ddot{\theta}$ (variable `f4`) includes the torque generated by suspension forces, the geometric forces, and the inertial load transfer due to the height of the center of gravity.

The total longitudinal force acting on the vehicle is $F_{tot\_x} = F_{x,f} + F_{x,r}$. This force creates a pitching moment around the center of gravity defined by $M_{long} = F_{tot\_x} \cdot h_{cg}$. The modified rotational equation of motion is:

$$J\ddot{\theta} = d_f \left(F_{sf} + F_{af} + F_{v,f}^{geo}\right) - d_r \left(F_{sr} + F_{ar} + F_{v,r}^{geo}\right) + F_{tot\_x} \cdot h_{cg} \tag{3.8}$$

Here, positive F_x (acceleration) creates a positive pitch moment (nose up), which is counteracted by the Anti-Squat force $F_{v,r}^{geo}$. Conversely, negative F_x (braking) creates a negative pitch moment (nose down), counteracted by the Anti-Dive force $F_{v,f}^{geo}$.

3.4.3 Impact on Linearization

Since the longitudinal forces $F_{x,f}$ and $F_{x,r}$ are treated as external inputs (or disturbances depending on the control architecture), the system matrices B and D in the state-space representation are updated. Specifically, the disturbance matrix regarding longitudinal inputs reflects that a change in throttle or braking now instantaneously affects the vertical acceleration $\ddot{z}$ and pitch acceleration $\ddot{\theta}$ through the γ coefficients and h_{cg}.

3.5 Control Strategy: LQR with Integral Action (LQR-PID)

To achieve precise tracking of the reference height and pitch angle while maintaining passenger comfort, a Linear Quadratic Regulator (LQR) with integral action is implemented. This approach effectively combines the optimal state feedback of LQR with the zero steady-state error property of a PID-type controller.

3.5.1 Augmented System State-Space Representation

In the MATLAB implementation, the control is designed for the controllable part of the system (the 8 physical states of the vehicle body and wheels). To eliminate steady-state offsets in heave (z) and pitch (θ), the state vector is augmented with two integral states, $e_{int,z}$ and $e_{int,\theta}$, defined as the integral of the error between the measured outputs and the references:

$$x_{aug} = \begin{bmatrix} x_{phys} \\ e_{int} \end{bmatrix} \in \mathbb{R}^{10\times 1}, \quad \text{where } e_{int} = \int (y_{ref} - y)\, dt \tag{3.9}$$

The augmented system matrices, A_{aug} and B_{aug}, are constructed as:

$$A_{aug} = \begin{bmatrix} A_{ctrl} & \mathbf{0}_{8\times 2} \\ C_{E,ctrl} & \mathbf{0}_{2\times 2} \end{bmatrix}, \quad B_{aug} = \begin{bmatrix} B_{ctrl} \\ D_{E1,ctrl} \end{bmatrix} \tag{3.10}$$

where $C_{E,ctrl}$ and $D_{E1,ctrl}$ map the physical states to the tracking errors for heave and pitch.

3.5.2 Optimal Control Law and Tuning

The control law is defined as $u = -K_{aug}x_{aug}$, where K_{aug} is the gain matrix that minimizes the quadratic cost function:

$$J(u) = \int_0^\infty (x_{aug}^T Q_{aug} x_{aug} + u^T R_{aug} u)\, dt \tag{3.11}$$

The tuning is performed through the selection of the diagonal matrices Q_{aug} and R_{aug}. Specifically, the weights in the MATLAB script are organized to balance different performance requirements:

- **Proportional Weights (q_{pz}, q_{ptheta}):** Regulate the reactivity of the system to position errors.
- **Derivative Weights (q_{vz}, q_{vtheta}):** Control the damping of the vertical and angular velocities.
- **Integral Weights ($q_{heave\_int}, q_{pitch\_int}$):** Ensure the elimination of steady-state errors caused by constant disturbances or load changes.

3.5.3 State Estimation: Kalman Filter

Since not all states are directly measurable, a Kalman Filter is designed to estimate the 12 physical states x. The observer gain K_o is calculated by solving the Riccati equation using the process noise covariance Q_{kalman} and the measurement noise covariance R_{kalman}:

$$\dot{\hat{x}} = A_N\hat{x} + B_1u + K_o(y - C_N\hat{x} - D_1u) \tag{3.12}$$

The observer allows the controller to use estimated velocities and road states that are otherwise inaccessible via standard sensors.

3.6 Numerical Control Design Analysis

Based on the theoretical framework established in the previous sections, the numerical synthesis of the control law was performed. The resulting matrices confirm the design choices aimed at achieving superior ride comfort ("Magic Carpet" effect) and robust handling.

3.6.1 Augmented System Dynamics

The augmented state matrix $\mathcal{A}_{aug} \in \mathbb{R}^{10\times 10}$, which includes the integral states for heave and pitch error regulation, is derived as:

Matrix $\mathbf{A_{aug}}$ (scaled by 10^3):

$$\begin{bmatrix} 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -27.5 & -3.3 & 0 & 0 & 13.7 & 1.6 & 13.7 & 1.6 & 0 & 0 \\ 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -47.5 & -5.7 & 14.4 & 1.7 & -14.4 & -1.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 729.2 & 87.5 & 1203.1 & 144.4 & -6354.2 & -87.5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 & 0 \\ 729.2 & 87.5 & -1203.1 & -144.4 & 0 & 0 & -6354.2 & -87.5 & 0 & 0 \\ 1.0 & 0 & 0 & 0 & -0.5 & 0 & -0.5 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.13}$$

The last two rows of $\mathcal{A}_{aug}$ (indices 9 and 10) correspond to the integral action dynamics. The terms 1.0×10^{-3} (scaled by 10^3 to unity) represent the integration of the vertical position error e_z and pitch error e_θ, ensuring zero steady-state error.

The input matrix $\mathcal{B}_{aug} \in \mathbb{R}^{10\times 2}$ reflects the direct influence of the control inputs u_1 (heave force) and u_2 (pitch moment) on the vertical velocity (state 2) and pitch rate (state 4), as well as on the unsprung mass dynamics. **Matrix $\mathbf{B_{aug}}$**:

$$\begin{bmatrix} 0 & 0 \\ 0.0004 & 0 \\ 0 & 0 \\ 0 & 0.0002 \\ 0 & 0 \\ -0.0104 & -0.0063 \\ 0 & 0 \\ -0.0104 & 0.0063 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.14}$$

3.6.2 Optimal LQR Gains Analysis

The control law $u(t) = -K_{prop}\hat{x}(t) - K_{int}\eta(t)$ is characterized by the computed gain matrices.

Proportional-Derivative Action (K_{prop})

The proportional and derivative gains for the physical states are given by:

$$K_{prop} = 10^4 \cdot \begin{bmatrix} 1.67 & 0.54 & 0 & 0 & -0.80 & 0.001 & -0.80 & 0.001 \\ 0 & 0 & 8.93 & 1.52 & -2.28 & 0.004 & 2.28 & -0.004 \end{bmatrix} \tag{3.15}$$

Analysis of these values highlights the control strategy:

- **Pitch Priority:** The gain on the pitch angle ($8.93 \cdot 10^4$) is significantly larger than the gain on vertical displacement ($1.67 \cdot 10^4$). This reflects the design intent to heavily suppress pitch motions, which are critical for passenger comfort in a luxury vehicle.
- **Active Damping:** The derivative gains (columns 2 and 4: $0.54 \cdot 10^4$ and $1.52 \cdot 10^4$) provide substantial active damping, replacing passive dissipation to quickly attenuate oscillations.
- **Sky-Hook Approximation:** The gains on unsprung mass velocities (columns 6 and 8) are negligible (≈ 0), indicating a control strategy that approximates a "Sky-Hook" damper, isolating the body from high-frequency wheel motions.

Integral Action (K_{int})

The integral gains are:

$$K_{int} = 10^5 \cdot \begin{bmatrix} 0.22 & 0 \\ 0 & 1.58 \end{bmatrix} \tag{3.16}$$

The integral gain for pitch ($1.58 \cdot 10^5$) is roughly an order of magnitude higher than for heave ($0.22 \cdot 10^5$). This aggressive integral action on pitch ensures rapid correction of static attitude changes during acceleration or braking (anti-squat/anti-dive compensation).

3.6.3 State Estimation and Road Profile Reconstruction

The full-order Kalman Observer gain matrix $K_o \in \mathbb{R}^{12\times 5}$ (reported in the Appendix) demonstrates the observer's bandwidth. Notably, the gains associated with the road profile states (rows 9-12 of K_o) exhibit values up to $2.0 \cdot 10^4$. This high sensitivity to sensor innovations ($y - \hat{y}$) confirms that the observer is tuned to rapidly estimate road irregularities ("epsilon" states), effectively distinguishing them from sensor noise. This allows the controller to react to road bumps almost instantaneously, approximating a feed-forward response.

Full Kalman Gain $\mathbf{K_o}$ (10^5):

$$\begin{bmatrix} -0.00 & -0.32 & -0.00 & -0.16 & -0.16 \\ -0.00 & -0.07 & -0.00 & -0.22 & -0.22 \\ 0.31 & -0.00 & 0.00 & 0.05 & -0.05 \\ 0.00 & 0.00 & 0.32 & -0.00 & 0.00 \\ 0.05 & 0.06 & 0.00 & -0.37 & -0.07 \\ 0.71 & 0.03 & 8.31 & -2.12 & 2.51 \\ -0.05 & 0.06 & -0.00 & -0.07 & -0.37 \\ -0.71 & 0.03 & -8.31 & 2.51 & -2.12 \\ -0.00 & 0.00 & 0.00 & 0.00 & -0.00 \\ 0.00 & -0.00 & -0.00 & -0.00 & 0.00 \\ 0.02 & 0.00 & 0.21 & -0.06 & 0.06 \\ -0.02 & 0.00 & -0.21 & 0.06 & -0.06 \end{bmatrix} \tag{3.17}$$

Chapter 4

Simulation Experiments and Results

This section presents a detailed discussion of the simulation experiments conducted on the half-car model. The experiments are designed to illustrate the behavior of the nonlinear plant, the linearized approximation, and the effect of control strategies including Anti-Dive/Anti-Squat geometry and integral action.

All simulations were performed using the MATLAB/Simulink implementation described in the previous sections. Plots show time histories of heave displacement, pitch angle, and relevant accelerations.

4.0.1 Open-Loop Linear vs Nonlinear Response

As a first step, we consider a baseline scenario without any control action. The vehicle is subjected to a longitudinal acceleration and deceleration profile, applied through front and rear wheel forces, while the road is assumed flat.

The Force is 12,25kN, all on the rear in acceleration while for the braking we chose a 60/40 division between front and rear tyre.

Figure 4.1 shows the vertical acceleration and pitch response of both the nonlinear plant and its linear approximation.

Several observations can be made:

- The linearized model approximates the nonlinear dynamics very closely in terms of amplitude and timing, validating the accuracy of the Jacobian-based linearization.
- However, the pitch angle reaches a peak of approximately 0.06 rad in the nonlinear model, which is undesirably high and can be uncomfortable for passengers.
- The mismatch is mostly visible during the initial transient, while steady-state deflections are similar.

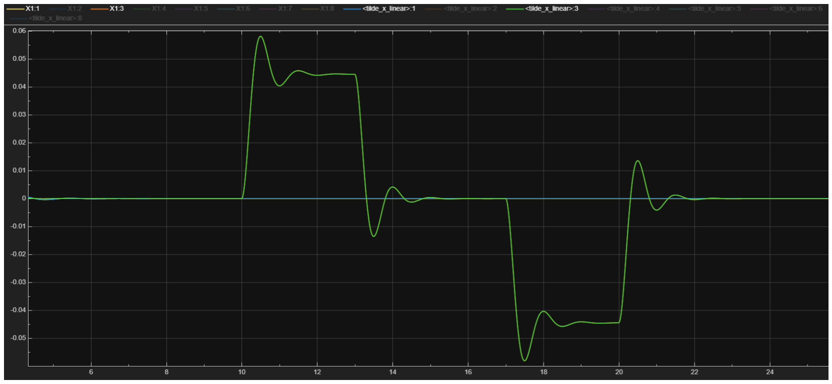


Figure 4.1: Open-loop response: vertical acceleration and pitch angle. Linear vs nonlinear model.

This experiment establishes the baseline performance and demonstrates the need for targeted pitch control.

4.0.2 Effect of Anti-Dive and Anti-Squat Geometry

Next, we enable the geometric load transfer effects represented by the Anti-Dive (γ_f) and Anti-Squat (γ_r) coefficients. This modification redistributes vertical forces during acceleration and braking to partially counteract pitch motion.

Figure 4.2 shows the resulting pitch angle and vertical displacement time histories.

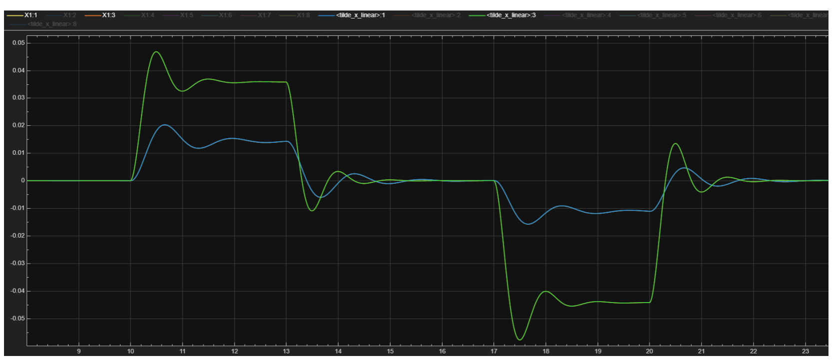


Figure 4.2: Vehicle response with Anti-Dive and Anti-Squat geometry. Pitch is slightly improved but vertical displacement oscillates more.

Key observations:

- The peak pitch angle is slightly reduced, showing that anti-dive and anti-squat provide some passive mitigation.

- However, the vertical heave varies significantly, particularly during braking or acceleration peaks. This occurs because the geometric forces redistribute load in a way that reduces pitch but does not increase overall suspension damping; effectively, the vertical stiffness of the system is partially bypassed, leading to higher p_z excursions.

- These results highlight a classic trade-off: purely geometric strategies can help with one degree of freedom (pitch) but may negatively impact another (heave).

4.0.3 Road-Induced Oscillations without Anti-Dive / Anti-Squat

In a further experiment, the Anti-Dive and Anti-Squat effects are removed and the vehicle is equipped qith the observer. It is clear how the control in this case has a much better response than a physical approach like the Anti-Dive and Anti-Squat configuration. We spent some time trying out different weights to find the better setup, during this we saw how much good vs bad weights can make the difference.
In the end we reached an improvement of one order of magnitude in angle.

Figure 4.3 shows the comparison between the nonlinear model without controls and the linear model with the observer

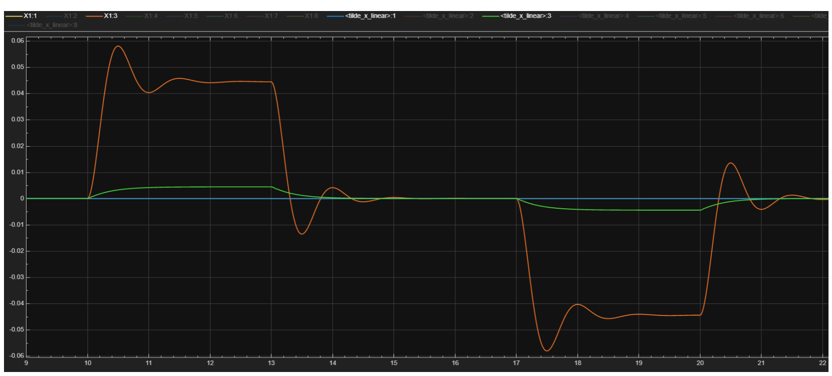


Figure 4.3: Observer improvement

Key points:

- Transient peaks are totally removed

- Pitch motion shows a dramatic reduction compared to previous cases, with an order-of-magnitude improvement in peak θ.

- Vertical movement is null, contrary to the result of Anti-Dive and Anti-Squat

This demonstrates that properly designed dynamic inputs can excite desirable modes while mitigating pitch without relying solely on geometric compensation.

4.0.4 Integral Action and Trade-Offs

Finally, we examine the effect of adding integral action on the heave and pitch errors, as implemented in the augmented LQR controller.

Figure 4.4 shows the heave and pitch response with integral action enabled.

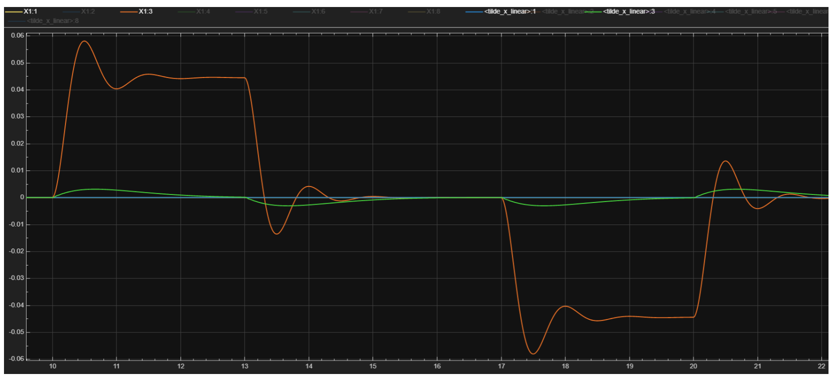


Figure 4.4: Vehicle response with integral action. Steady-state error reduced but transients slightly slower.

Observations:

- Integral action ensures that both heave and pitch converge to the desired equilibrium values (r_z and r_θ) in steady state.
- Transient performance is slightly degraded, with longer rise times and mild overshoot, reflecting the trade-off between zero steady-state error and transient responsiveness.
- This behavior highlights the classical control trade-off: integral action eliminates bias but can reduce damping or slow the system response.

4.0.5 Severe Road Bump Scenario

In order to further stress the model and controller, a severe road disturbance is introduced in the form of a sinusoidal bump. This test represents a highly challenging scenario for the suspension system, as it excites both heave and pitch modes simultaneously with significant amplitude and high vertical acceleration content.

The bump profile is defined as a half-sine wave with:

- Height: 0.20 m,

- Length: 0.20 m,
- Vehicle speed: approximately 20 km/h.

This configuration produces a short-duration, high-curvature excitation, corresponding to a very demanding input for both passive and active suspension dynamics.

Case 5: Linear vs Nonlinear Model under Bump Excitation

We first compare the open-loop response of the nonlinear plant and its linear approximation.

Figure 4.5 shows the heave displacement and pitch angle for both models.

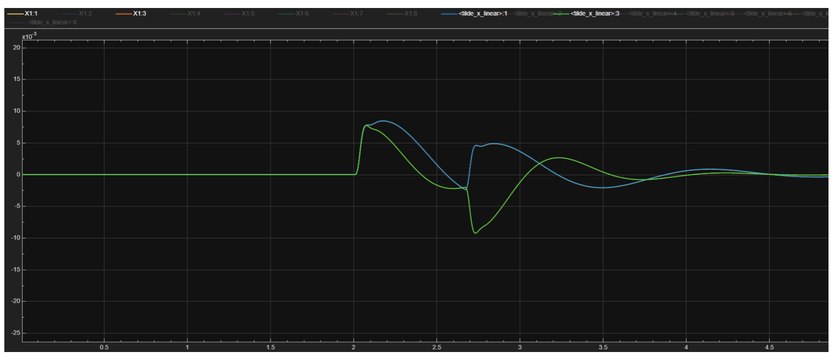


Figure 4.5: Open-loop bump response: nonlinear vs linear model.

Despite the severity of the excitation, several important observations can be made:

- The linear and nonlinear models match remarkably well, both in amplitude and phase.
- Peak pitch and heave responses are almost indistinguishable during the main transient.

This result is particularly significant because the bump amplitude (20 cm) and short wavelength (20 cm) introduce large suspension deflections and high vertical accelerations. The close agreement confirms that the linearized model remains valid even under aggressive excitation conditions, strengthening confidence in the controller design framework based on the linear model.

Case 6: Controlled Linear Model vs Uncontrolled Nonlinear Plant

Next we compare:

- The uncontrolled nonlinear plant,

- The controlled linear model with tuned LQR gains.

The controller was primarily tuned to mitigate pitch oscillations during longitudinal acceleration and braking, which represented the most critical scenario observed previously. No specific tuning was performed to directly minimize vertical heave displacement under sharp bump excitation.

Figure 4.6 shows the comparative results.

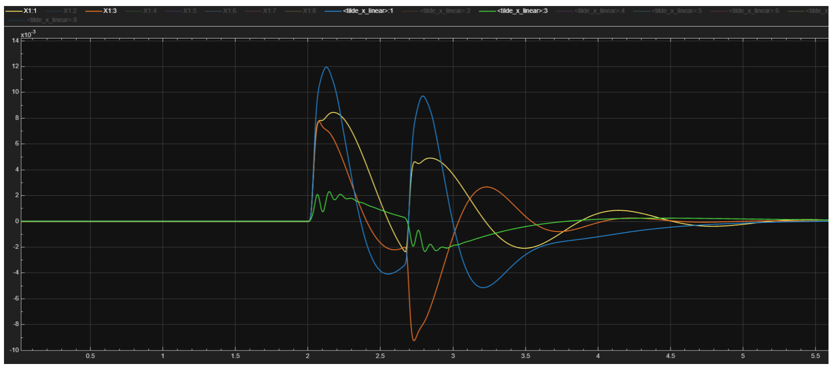


Figure 4.6: Comparison between uncontrolled nonlinear plant and controlled linear model under bump excitation.

The results reveal a clear trade-off:

Pitch Response

- The pitch angle is significantly improved in the controlled case.
- Peak θ is substantially reduced compared to the uncontrolled nonlinear response.
- Oscillations decay faster, demonstrating effective damping injection through the control law.

Although the controller was tuned primarily for acceleration-induced pitch, the improvement carries over to the bump scenario, confirming robustness of the design.

Heave Response

- The vertical displacement p_z is slightly worse in the controlled configuration.
- Peak heave excursion increases marginally compared to the uncontrolled case.

This behavior can be explained by the fact that the cost function weights were selected to strongly penalize pitch and angular velocity, rather than vertical displacement. As a consequence, the controller actively redistributes suspension forces to counteract rotational motion, sometimes at the expense of increased vertical travel.

In other words, the controller sacrifices a small amount of heave performance in order to achieve substantial pitch stabilization.

Sensor Interpretation and Measurement Limitations

An additional observation concerns the sensor signals used for feedback.

During fast vertical variations, such as those induced by the short sinusoidal bump:

- The vertical accelerometer signal becomes highly oscillatory.
- Gravity compensation and frame projection introduce additional sensitivity to noise and modeling mismatch.

In such cases, the accelerometer does not provide a clean measure of slow heave displacement, as it primarily captures high-frequency vertical acceleration components.

Conversely:

- The gyroscope signal remains clean and directly proportional to pitch rate.
- Rotational dynamics are therefore more reliably observable and controllable.

This explains why pitch control remains highly effective, while heave regulation is less precise under sharp vertical disturbances.

The controller therefore relies more heavily on gyroscopic information for stabilization in aggressive bump scenarios.

4.0.6 Discussion of Trade-Offs

The bump experiments highlight several fundamental control trade-offs:

1. The linear model remains an excellent approximation even under severe nonlinear excitation.
2. Control tuned for longitudinal load transfer scenarios (acceleration/braking) generalizes well to bump-induced pitch disturbances.
3. Improving pitch stabilization may slightly degrade vertical displacement performance.
4. Accelerometer-based feedback becomes less informative under rapid vertical oscillations, whereas gyroscope measurements retain high reliability.

Overall, the controller demonstrates strong robustness and significantly improves pitch comfort under aggressive road disturbances, even though vertical displacement was not the primary tuning objective.

4.0.7 Additional Experimental Scenarios

In order to further investigate the interaction between active control and suspension geometry, two additional experimental configurations were analyzed. These tests aim to clarify the influence of Anti-Dive and Anti-Squat geometry when combined with state estimation and integral action, as well as their effect under different longitudinal force distributions.

Case 7: Observer + Integral Control with Anti-Dive and Anti-Squat

In this experiment, the controller includes:

- State observer,
- Integral action on heave and pitch,
- Anti-Dive and Anti-Squat geometric coefficients enabled.

The objective is to evaluate whether geometric load transfer compensation improves closed-loop behavior when combined with active control.

Figure 4.7 illustrates the system response.

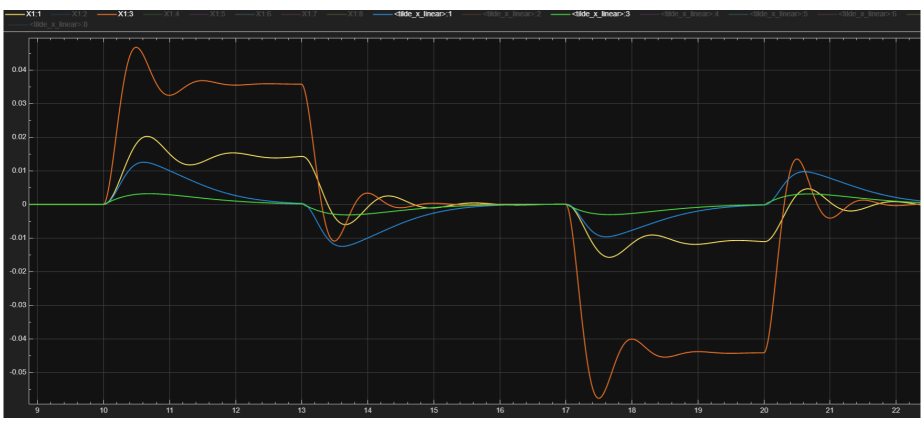


Figure 4.7: Case 7: Controlled system with observer, integrator, and Anti-Dive/Anti-Squat geometry.

Surprisingly, the performance is slightly worse compared to the configuration without Anti-Dive and Anti-Squat.

In particular:

- Vertical displacement exhibits larger excursions.

- Pitch stabilization is still really good.
- Transient behavior appears less damped.

This behavior can be explained by analyzing how geometric load transfer interacts with active control.

Anti-Dive and Anti-Squat modify the effective vertical force distribution during longitudinal acceleration. Instead of allowing suspension springs and dampers to absorb load transfer naturally, part of the longitudinal force is redirected directly into vertical reaction forces at the chassis.

From a control perspective, this introduces an additional coupling between longitudinal dynamics and heave motion. The controller, which was tuned assuming a certain structure of the plant, now faces a modified effective vertical stiffness and damping distribution.

Moreover:

- The integral action attempts to restore the nominal heave equilibrium.
- The geometry simultaneously injects vertical forces proportional to longitudinal acceleration.

These two mechanisms can partially conflict, leading to larger oscillations in p_z.

In other words, the geometric compensation modifies the equilibrium configuration dynamically, while the integrator attempts to enforce a fixed reference height. This mismatch results in reduced overall performance.

This experiment demonstrates that passive geometric compensation does not necessarily combine optimally with active closed-loop control.

Case 8: Four Acceleration Profiles with Different Force Distributions

In the final experiment, four longitudinal acceleration cases are considered. Each case maintains the same total longitudinal force but varies its distribution between front and rear wheels.

Anti-Dive and Anti-Squat geometry remain enabled.

Figure 4.8 shows the comparative pitch and heave responses.

The results clearly show that the response differs significantly between the four cases.

This effect is directly attributable to the geometry coefficients γ_f and γ_r.

Anti-Dive and Anti-Squat generate vertical forces proportional to the longitudinal tire forces:

$$F_{v,long,f} = \gamma_f F_{xf}, \qquad F_{v,long,r} = \gamma_r F_{xr}.$$

Therefore:

- If acceleration is front-dominated, the front suspension receives stronger vertical coupling.

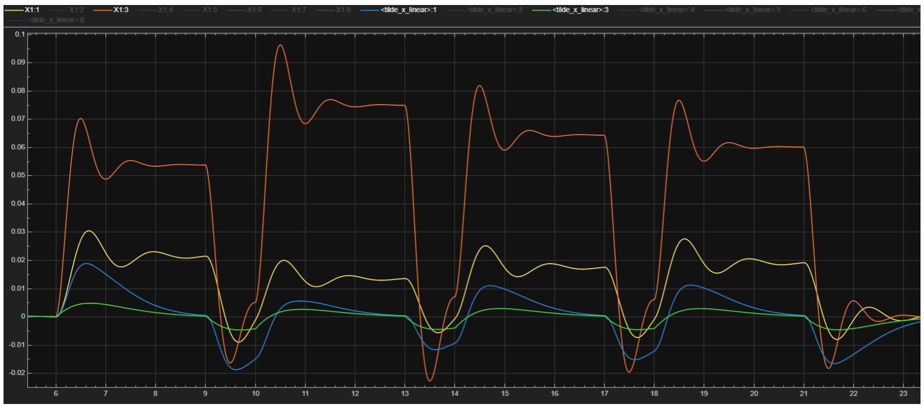


Figure 4.8: Case 8: Four acceleration cases with different front/rear force distributions under Anti-Dive and Anti-Squat geometry.

- If rear-dominated, the rear suspension is more affected.

As a consequence, different front/rear torque distributions produce distinct pitch and heave behaviors even when the total longitudinal acceleration is identical.

Without Anti-Dive and Anti-Squat, these four cases would produce nearly identical global heave behavior, since only total longitudinal force would matter. With geometric compensation enabled, however, the system becomes sensitive to the distribution of forces.

This experiment highlights an important design insight:

Anti-Dive and Anti-Squat introduce distribution-dependent behavior, making the vehicle response sensitive to torque vectoring or drivetrain configuration.

From a control design perspective, this implies that:

- Controllers must account for force distribution if geometric compensation is present.
- Alternatively, removing geometric load transfer simplifies the plant and yields more predictable closed-loop behavior.

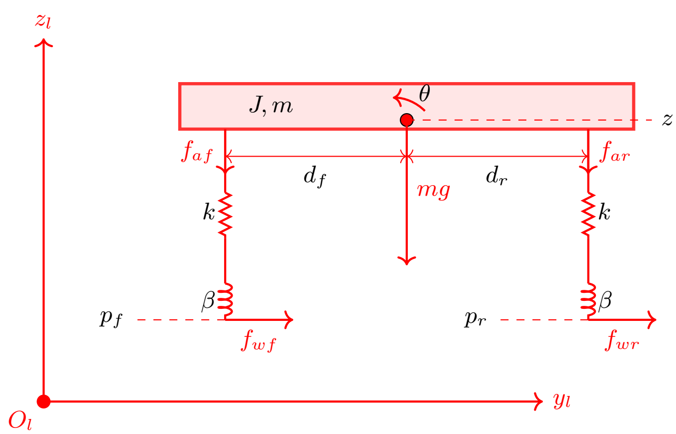
4.0.8 Overall Insight from Additional Experiments

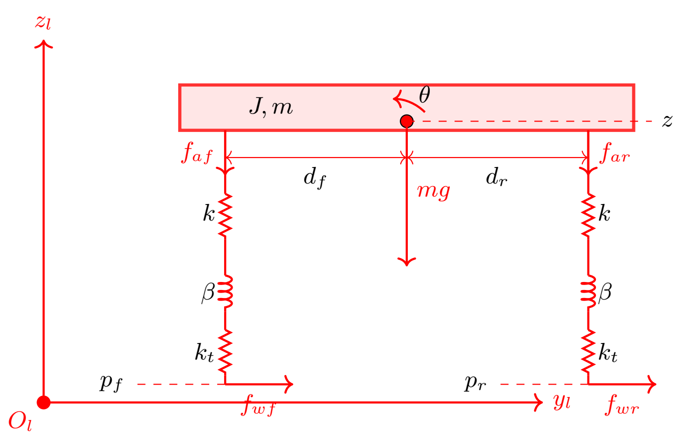
The additional experiments confirm that:

1. Anti-Dive and Anti-Squat can conflict with integral action in closed-loop control.
2. Geometric load transfer introduces additional coupling between longitudinal acceleration and vertical equilibrium.

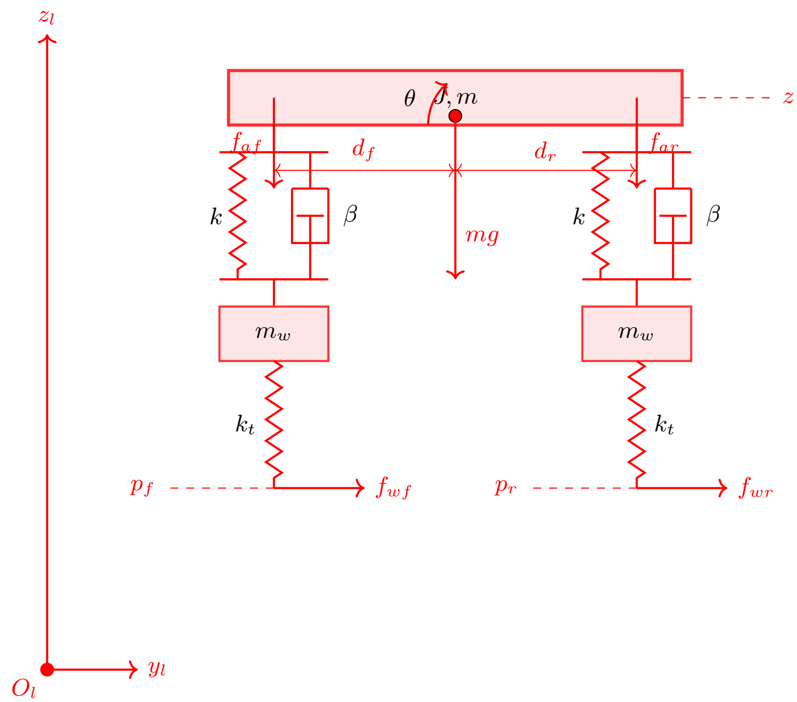
3. Different front/rear acceleration distributions lead to distinct vertical responses when geometric compensation is active.

These findings reinforce the importance of carefully balancing passive suspension geometry and active control strategies when designing advanced ride control systems.





todo: finire sto disegno



Copy and past the Simulink block scheme and describe what each block does. Describe the set-up MATLAB file, where and how to change the parameters of the simulations. Remember to include also the sensor noises and realistic external disturbances.

4.1 Simulation results

Describe the simulation scenario: initial conditions, purpose of the simulation. Describe the results: are the results coherent with the expectation? If not why? Investigate the tuning: how the performance are affected by the selection of the parameters at disposal of the designer?

Chapter 5

Conclusions and further investigation

Recap the main results obtained in the project and highlight eventual further investigation directions alogn which the performance could be improved.

Bibliography

List the papers/books cited.

Appendix

Use appendices to add technical parts which are instrumental for the completeness of the manuscript but are too heavy to be included inside the main text. Basically, appendices are exploited to let the main text cleaner and smoother. As example, the complete MATLAB listings can be reported in appendix.