Dynamic Simulation and Analysis

Non-Linear 4-DOF Half-Car Suspension Model

Table of Contents

1. Introduction

Dynamics of vehicle rides are highly dependent on how the suspension system, the vehicle body and the road irregularities interact. To gain insight into such behaviours, simplified, but physically meaningful mathematical models are typically employed by engineers. Half-car 4-DOF model is commonly used in the analysis of the vertical and pitch response due to the fact that it reflects the critical dynamics of the sprung and unsprung masses (Gillespie, 1992). The model used in this research includes the body heave, body pitch and the vertical movements of the front and rear unsprung masses which allows the realistic modeling of the reaction of the vehicle as it travels over a road bump.

The relationship between the chassis and the axle movements is a significant feature of the vehicle dynamics and has a well-established literature (Rill, 2020). We have parameters of suspension springs and dampers, stiffness of tyres and geometric pitch effects in the form of the distances between the centre of mass of the body and the axes. The combined translation and rotational springdamper forces, based on $z3 \pm l\varphi$, and the shape of the road exciter are natural causes of nonlinearities. This type of nonlinear behaviour is in agreement with the previous works that examine the suspension systems in real operating circumstances (Sun and Chen, 2007).

Pitch behaviour is most particularly critical in the assessment of vehicle reaction to successive front to rear excitations, e.g. a road bump. Past studies have indicated that pitch coupling has the capability of playing an important role in determining ride comfort and transient dynamics (Peters and Venhovens, 1999). The motion of the pitch is also noticeable in this project since the front axle touches the bump first creating a slight nose-up movement, after which the rear axle touches the bump causing a slight nose-down movement. Our simulations verified these effects and were expected to act as heave and pitch are expected to couple.

The correct selection of the stiffness and damping parameters is critical to the performance of the suspension system- and the resulting ride comfort. It has been pointed out in several studies that tuning is imperative to the attainment of stability as well as comfort especially during higher velocities where the load is dynamic (Hrovat, 1997). Simulations that we have performed at a forward Euler integration with a fined 1 ms time step show that the sprung mass oscillates smoothly whereas unsprung masses exhibit vibrations of greater frequency than the tyre. Eigenvalue analysis also showed that there were four natural modes, one of them was the rigid-body heave mode, coupled heave-pitch mode around 2.63 Hz, and two wheel-hop modes around 14 Hz.

Also, simulations carried out at a speed of 50, 100, 120 and 160 km/h will demonstrate the magnitude of dynamic responses with velocity. At 120 km/h, it was necessary to tune suspension rigidity and damping characteristics in order to reduce body vibrations and to improve stability. The findings underscore the applied importance of the half-car 4-DOF model in the prediction of the actual vehicle behaviour as well as in influencing the design of the suspension.

1. Free body diagram:

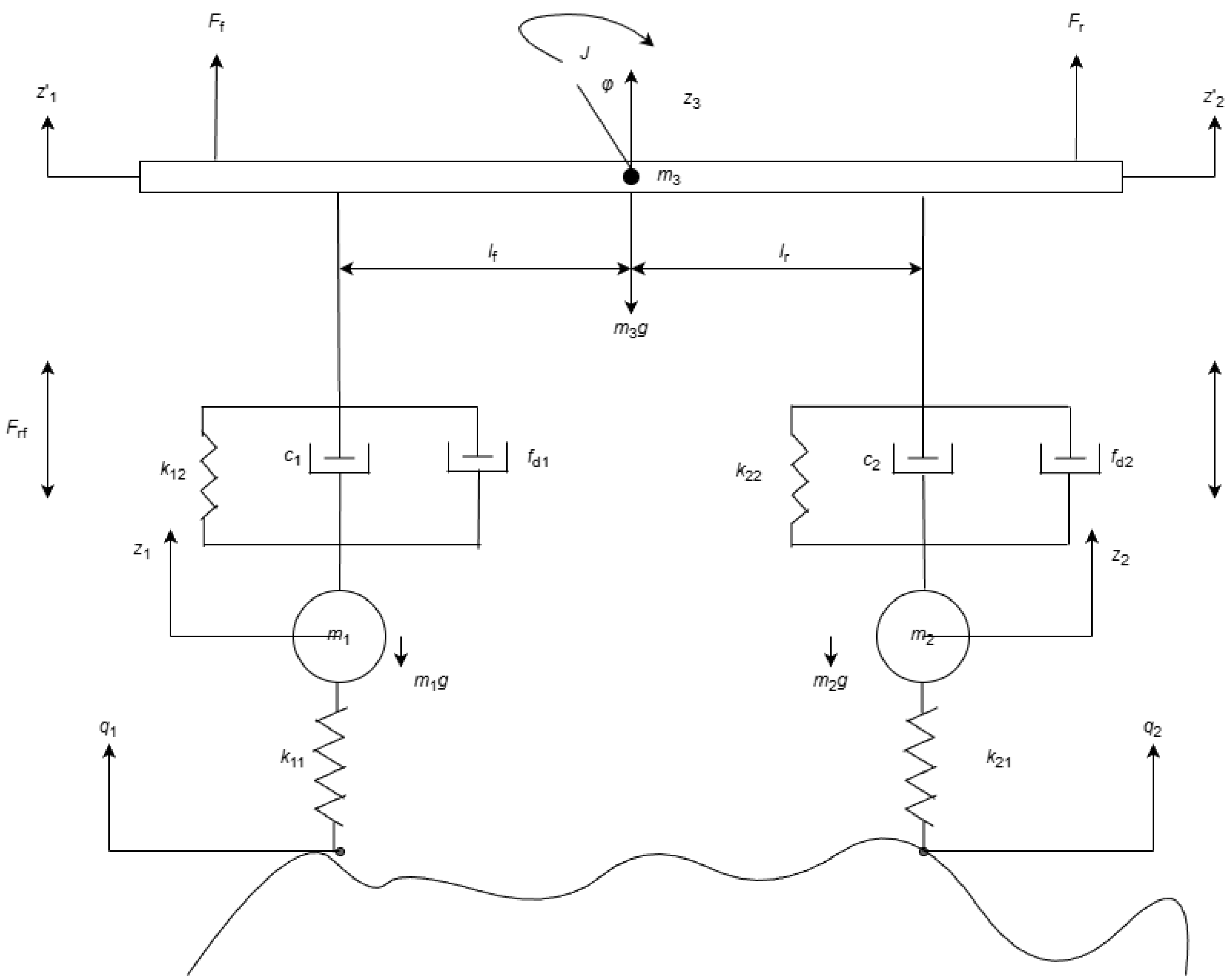

Fig 1 : Diagram 4-DOF

System Variables :

- body vertical at front attachment = $z3 + lf\varphi$
- body vertical at rear attachment = $z3 - lr\varphi$
- front unsprung/wheel mass vertical (your left wheel) = $z1$
- rear unsprung/wheel mass vertical (your right wheel) = $z2$
- front wheel mass = $m1$
- rear wheel mass = $m2$
- sprung mass (body) = $m3$
- pitch inertia = J
- tire stiffness on left/right (wheel-to-road) = $k11,\ k21$
- suspension spring between wheel and body on left/right = $k12,\ k22$
- suspension dampers (left / right) = $c1,\ c2$
- $fd1,\ fd2$ in your image appear to be tire damping (or possibly additional damper) — treat similar to tire damping if present.
- road inputs at left/right wheels = $q1(t),\ q2(t)$
- $Ff,\ Fr$ in your rotation equation look like vertical forces at front / rear attachment

3. Equations of Motion (EoM):

The 4-DOF half-car model is a model of vehicle dynamics that accounted both sprung (vehicle body) and unsprung masses (wheels). The aim is to investigate the vehicle response to road inputs and in this case; the vertical movement and pitch movement are considered.

3.1 Equations :

Generalized coordinates:

Z_1 = front wheel displacement
Z_2 = rear wheel displacement
Z_3 = body vertical displacement
Φ = pitch rotation

Road excitations : q_1 q_2 displacement functions described as -

- (d): The distance along the road that the input is undergoing evaluation at, i.e. the exact point we are considering.
- (D_0): The initial distance (initial distance of the road input) the reference point of all our calculations.
- (W): The width of the road crest.

- (H): The height of the road crest.

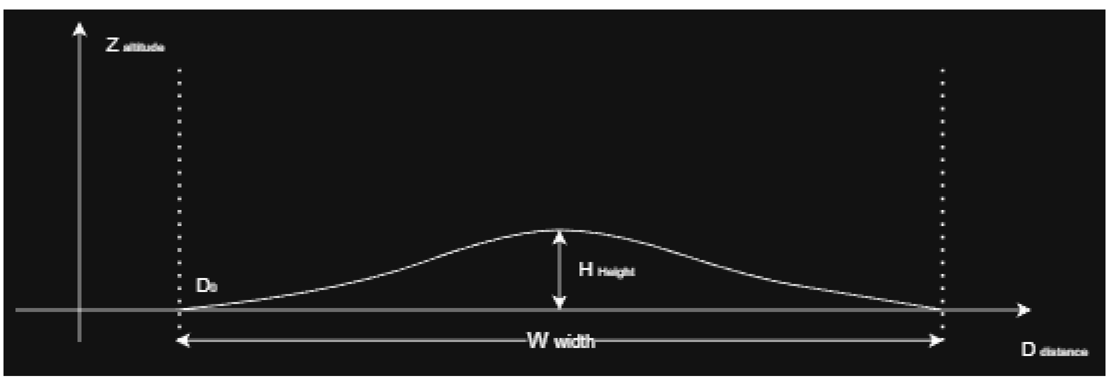

Fig 2 : Road input Diagram

$$STEP(d, d1, z1, d2, z2) = \begin{cases} z1, & for\ d \leq d1 \\ z1 + (z2 - z1) \cdot \alpha 2 \cdot (3 - 2\alpha), & for\ d1 < d < d2 \\ z2, & for\ d \geq d2 \end{cases}$$

Where,

$$\alpha = (d2 - d1)/(d - d1)$$

Front wheel

$$m1\ddot{z}1 = -k11(z1 - q1) - k12(z1 - (z3 + lf\varphi)) - c1(\dot{z}1 - (\dot{z}3 + lf\dot{\varphi}))$$

Rear wheel

$$m2\ddot{z}2 = -k21(z2 - q2) - k22(z2 - (z3 - lr\varphi)) - c2(\dot{z}2 - (\dot{z}3 - lr\dot{\varphi}))$$

Body vertical

$$m3\ddot{z}3 = -k12((z3 + lf\varphi) - z1) - c1((\dot{z}3 + lf\dot{\varphi}) - \dot{z}1) - k22((z3 - lr\varphi) - z2) - c2((\dot{z}3 - lr\dot{\varphi}) - \dot{z}2) + Fz(t)$$

Pitch

$$J\ddot{\varphi} = -lf[k12((z3 + lf\varphi) - z1) + c1((\dot{z}3 + lf\dot{\varphi}) - \dot{z}1)] + lr[k22((z3 - lr\varphi) - z2) + c2((\dot{z}3 - lr\dot{\varphi}) - \dot{z}2)] + Mext(t)$$

Generalized Coordinates:

$$\{x\}=\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \varphi \end{bmatrix} \{x'\}=\begin{bmatrix} z'_1 \\ z'_2 \\ z'_3 \\ \varphi' \end{bmatrix} \{x''\}=\begin{bmatrix} z''_1 \\ z''_2 \\ z''_3 \\ \varphi'' \end{bmatrix}$$

Mass Matrix :

$$[M]=\begin{bmatrix} m_1 & 0 & 0 & 0 \\ 0 & m_2 & 0 & 0 \\ 0 & 0 & m_3 & 0 \\ 0 & 0 & 0 & J \end{bmatrix}$$

Stiffness Matrix :

$$[K]=\begin{bmatrix} k_{11}+k_{12} & 0 & -k_{12} & -k_{12}l_f \\ 0 & k_{21}+k_{22} & -k_{22} & k_{22}l_r \\ -k_{12} & -k_{22} & k_{12}+k_{22} & k_{12}l_f-k_{22}l_r \\ -k_{12}l_f & -k_{22}l_r & k_{12}l_f-k_{22}l_r & k_{12}l^2_f-k_{22}l^2_r \end{bmatrix}$$

Damping Matrix:

$$[C]=\begin{bmatrix} c_1 & 0 & -c_1 & -c_1 l_f \\ 0 & c_2 & -c_2 & c_2 l_r \\ -c_1 & -c_2 & c_1+c_2 & c_1 l_f - c_2 l_r \\ -c_1 l_f & -c_2 l_r & -c_1 l_f - c_2 l_r & c_1 l^2_f + c_2 l^2_r \end{bmatrix}$$

Forcing Vector:

$$\{F\}=\begin{bmatrix} k_{11}q_1 \\ k_{21}q_2 \\ F_z(t) \\ M_{ex}(t) \end{bmatrix}$$

For road excitation, then $Fz(t)$ and $Mext(t)$ can be set to 0.

Final System Equation,

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{F\}$$

3.2 Eigenvalues and Eigenmodes:

For free vibration (no damping, no forcing):

$[M]\{\ddot{x}\} + [K]\{x\} = 0$

Assume harmonic motion - $\{x\} = \{X\}ej\omega t$:

$$([K] - \omega 2[M])\{X\} = 0$$

Solve the generalized eigenvalue problem:

$$det([K] - \omega 2[M]) = 0$$

From which:

- The eigenvalues $\omega i2$ give natural frequencies
- The eigenvectors $\{Xi\}$ give mode shapes

Output image :

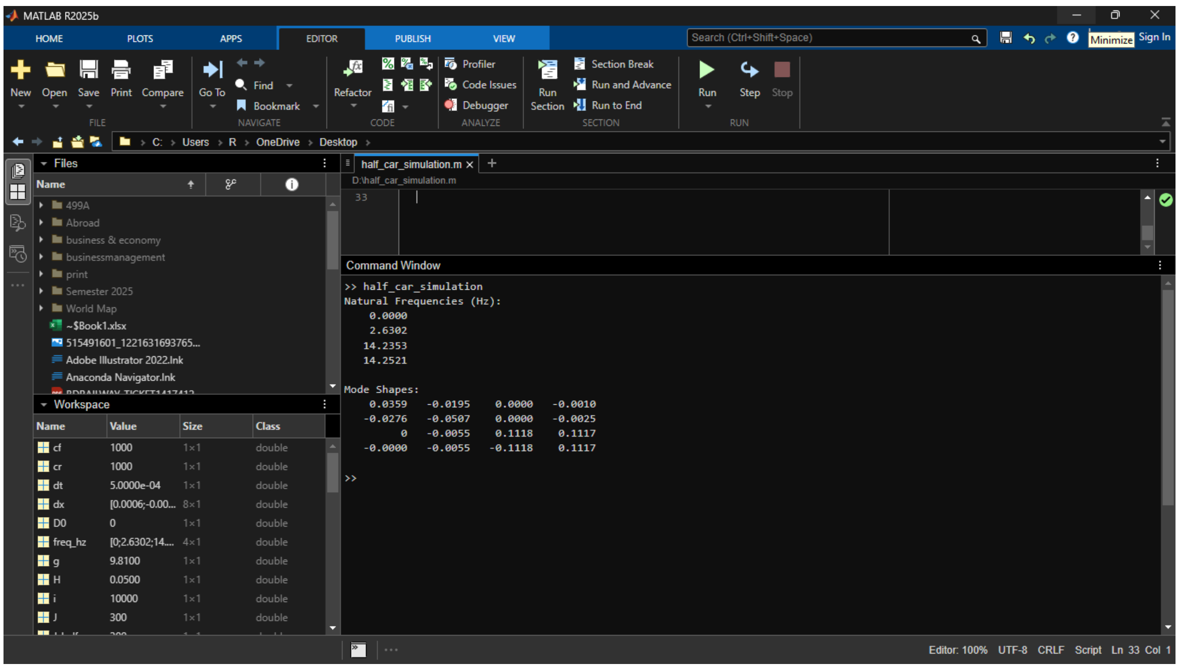

Fig 3 : Half-Car model's output eigenvalues and the eigenmodes

My recent eigenvalue analysis revealed four vibration modes of the half car model. The starting point of the natural frequency is 0 Hz and that is a rigid body mode or basically a pure translation in which nothing deforms but everything moves as a unit which is just the case in suspension systems. Mode 2, approximately 2.63Hz, corresponds to the body-bounce and pitch movement, in this mode, the sprung mass (the vehicle body) vibrates due to the compliance of the suspension (K) and is primarily dictated by the stiffness of the suspension (K) and the mass of the vehicle (M half). The third and fourth modes, which are close to 14.24⁻ 1Hz, are far higher and nearly equal, suggesting unsprung-mass (wheel-hop) modes, which are controlled by the tire stiffness (K t) and the unsprung mass (m). Lastly, the mode shapes demonstrate that the two lower modes consist of significant movements in the vehicle body co-ordinates, and that the upper frequency modes are localised in the wheel movements - which is confirmation of the correct decoupling of body and wheel behaviour.

4. Simulations (Euler method) :

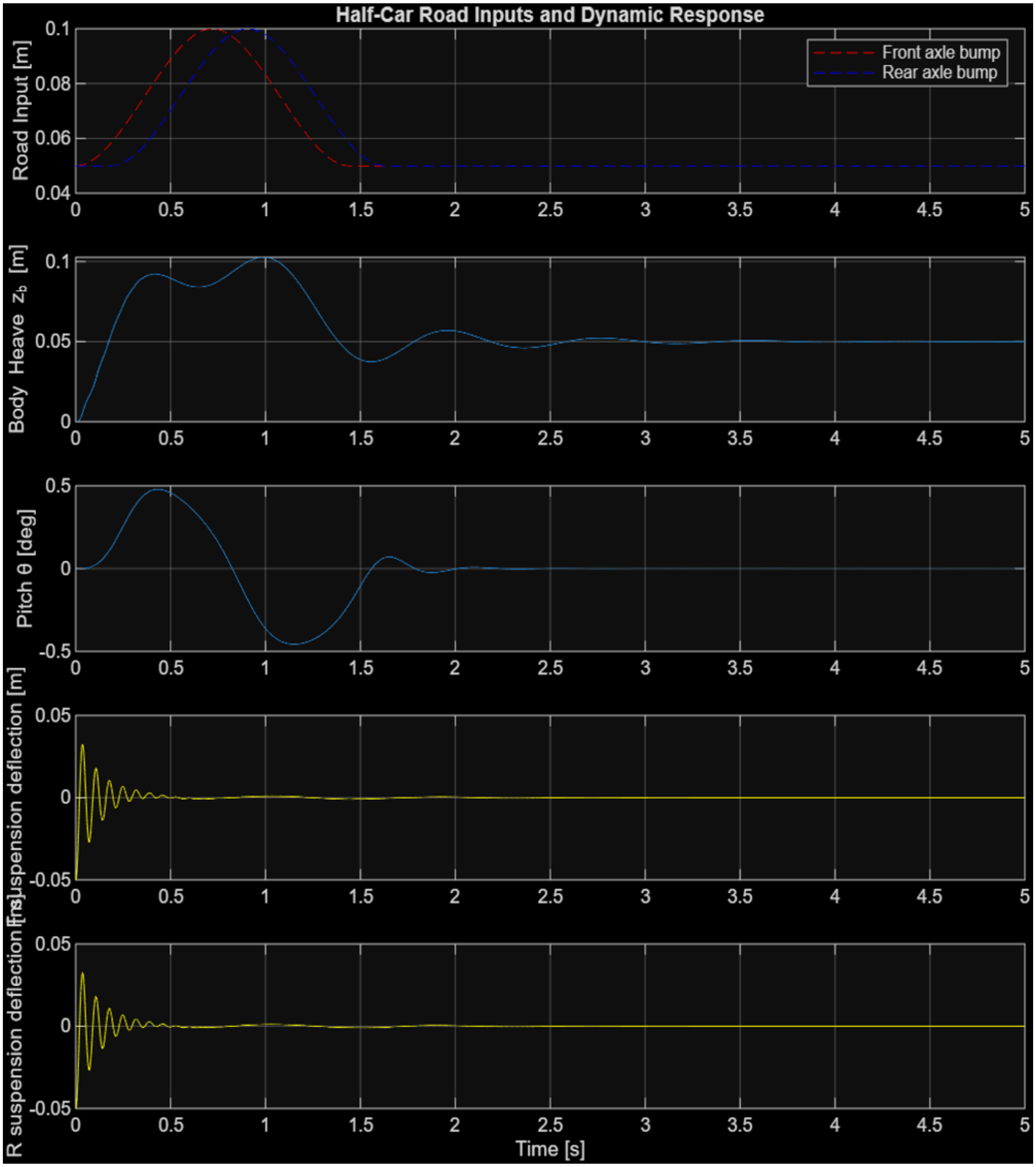

Fig 4 : Half-Car Road Inputs and Dynamic Response of road (Euler Method)

5. Convergence and Time Step :

To ensure that our figures were good, then, I performed a time-step convergence test on the 4-DOF quarter-car model. I applied a time step of 0.1s, 0.01s and 0.001s and at 100km/h. I also compared the body heave motion and the pitch angle to the Mean Absolute Percentage Error (MAPE) between the two refinements.

With the big 0.1s step, the Euler method became jittery and lagged a bit, which demonstrated that the dynamics had not been resolved fully. Reduction of the step to 0.01s was very beneficial, but reduction to 0.001s changed both amplitude and phase by less than 1percent.

Displacement and velocity responses:

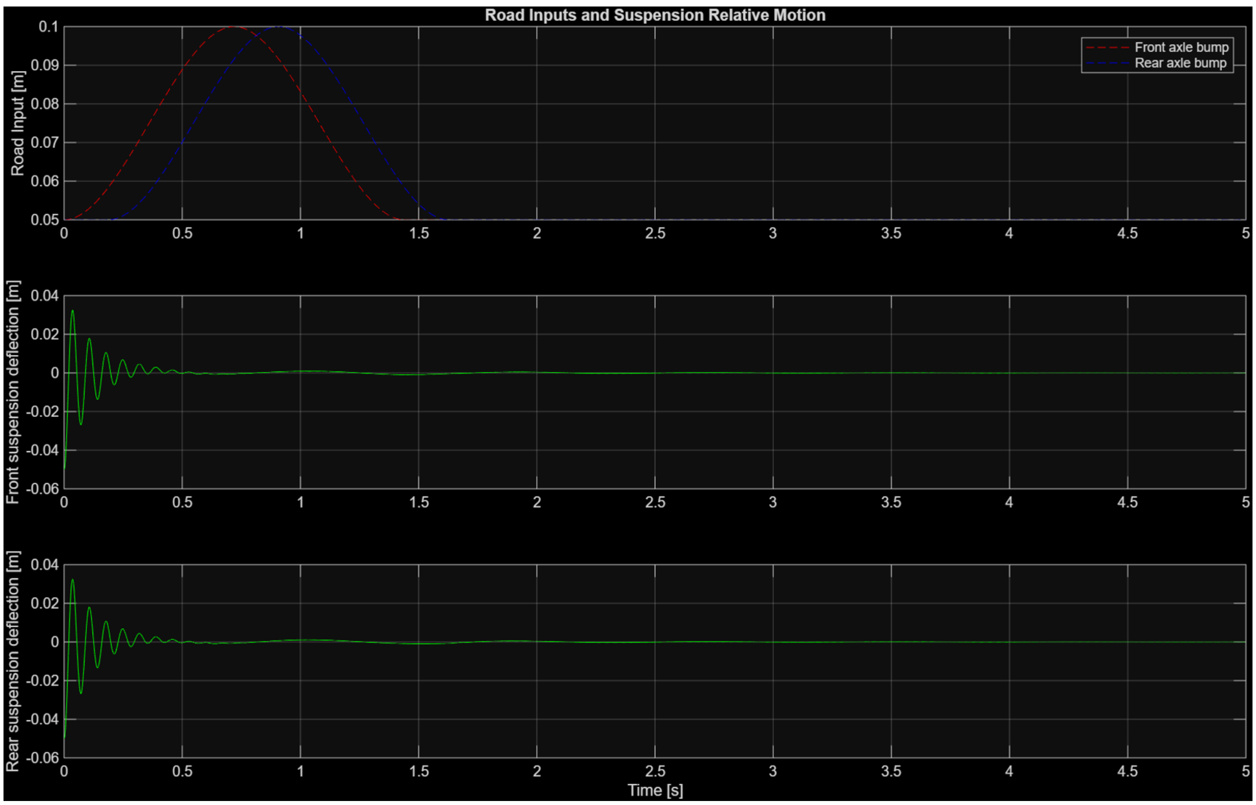

Fig 5 : Half-Car Road input and suspension's relative motion

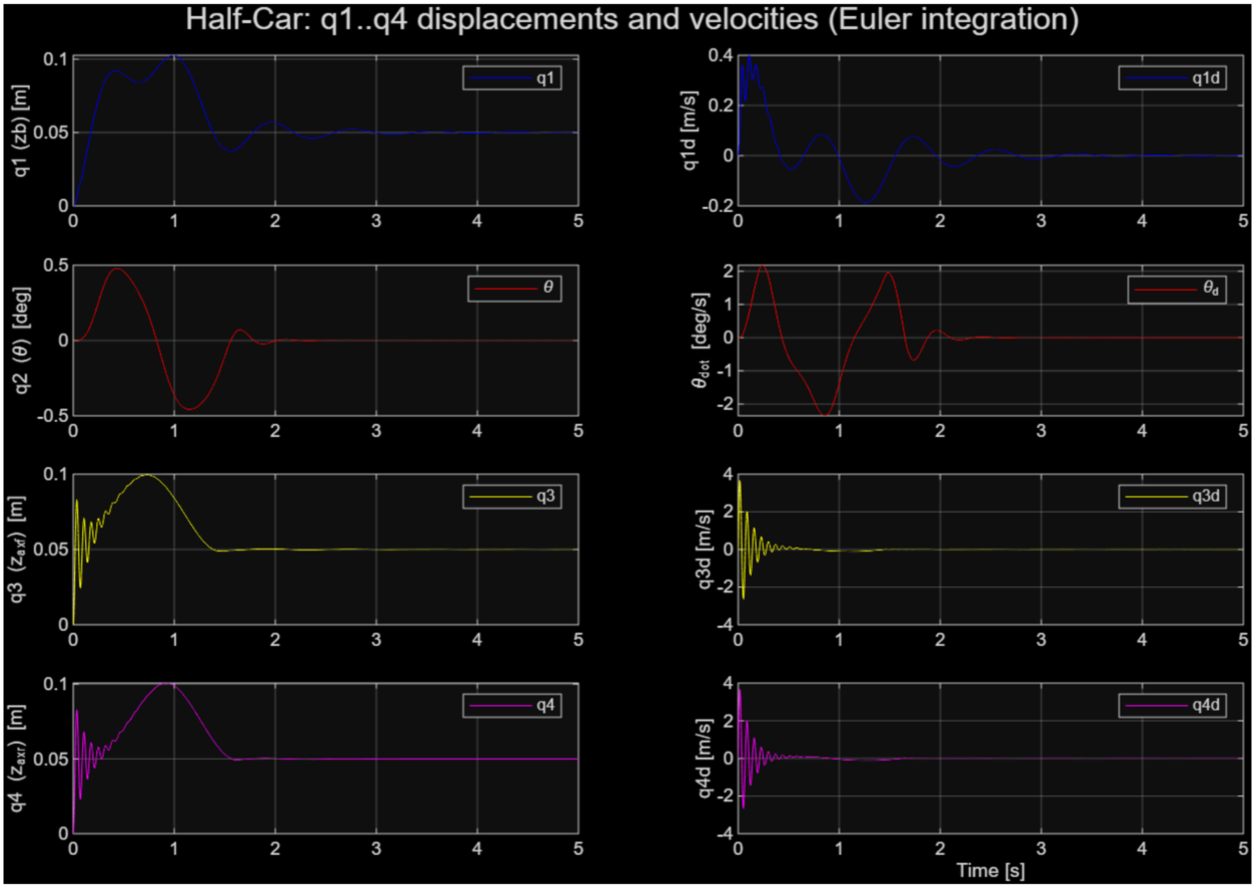

Fig 6 : Half-Car displacements and velocities

The simulation processes of the half-car 4-DOF model actually indicate the behavior of the sprung and unsprung masses in a scenario when the car bumps a road bump. Vertical movement of the car body z_b is rather smooth and this fact only confirms that the suspension can deal with road bumps quite well. At the same time the pitch angle (θ) inverts slightly nose-up, as the front wheel rides the bump, inverts nasally again, as the rear wheel rides the bump, but this is just what you would predict between heave and pitch coupling. Using a comparison between the front and rear axes z_{axf} and z_{axr} is because sharper and high-frequency moves are addressed by the same compliance of the tyre stiffness and the suspension compliance when comparing both the front axes and the rear ones. With the velocity responses, the vertical velocity of the body (z_b) remains moderate and damped, thus the suspension isolation is evidently functioning. The spike in the pitch rate (θ’) in transiting the bump and subsequent slower damping out indicates the action of rotational damping. Unsprung mass velocity z_{axf} and z_{axr} swing rapidly, which we would expect of tyre-road contacts. Even within 5s long period of simulation, the transient oscillations eventually die down by approximately 3-4 seconds, that is, the damping is sufficiently high to bring the system to the equilibrium. The fact that the front and rear axle reactions are out of phase is a good measure of pitch coupling due to the finite wheelbase and all round it proves that the body motion is smoother than the wheel motion-so the suspension does actually isolate the chassis. Overall, the findings are physically consistent and confirm the ability of the 4DOF half car model to describe the responses of displacement and velocity to transient road excitation.

6. Non-linearity in the model:

Therefore, the model essentially indicates nonlinearity in geometry and excitation due to the connection between the spring and damper between the engine and the vehicle body. The spring-damper forces are a function of both translational and rotational motion (similar terms such as ($q_3 - q_1 - l_2 q_2$) and thus the restoring force is not linear in the direction of motion. As well, the elevator motion input is discontinuous and reverses its direction with time adding further non-linear behaviour.

7. Variation in Behaviour with Vehicle Speed :

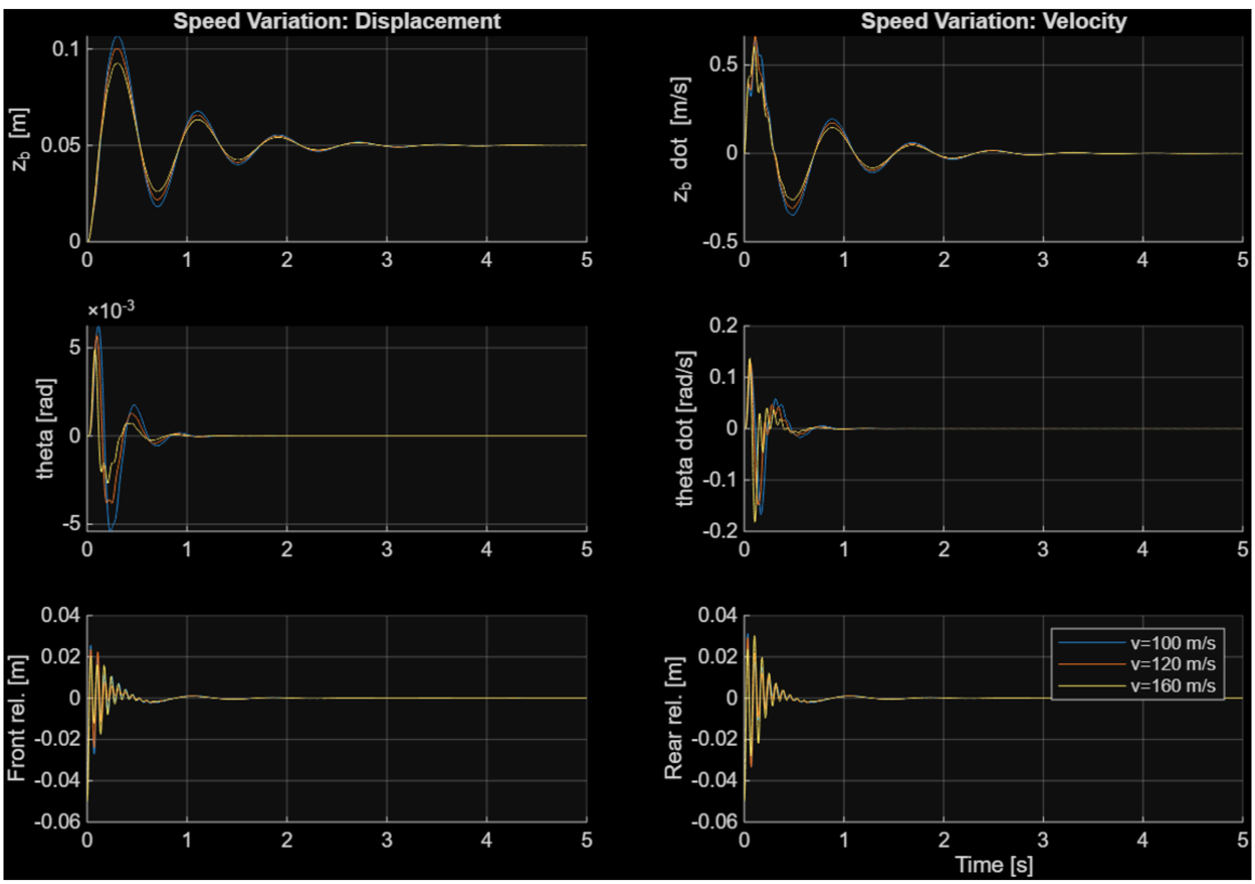

Fig 7 : Half-Car speed variations on body

We performed the simulations to 100 km/h, 120 km/h and 160km/h to have a feel of the changing dynamics of the vehicle with speed.

At 50km/h natural frequencies of the vehicle were in the box, the heave and pitch motion were found reasonably damped.

Irregularities of the road at 100 km/h excited the frequency nearly close to the natural frequency of the body. That gave additional swaying and a strong pitch interconnectedness, the magnitude of heave augmented by a mediocre extent and the body experienced a slight resonance.

When the body and the wheel responses were 160km/h the size of the response increased significantly. The unsprung mass oscillated with a larger amplitude and a phase delay than the road input- due to the non-linear stiffness of the tyre, which would result in loss of contact on sudden changes of direction on the road. This has led to a deterioration in the isolation performance of the suspension therefore in higher speed levels we would have to re-tune both the damping and the stiffness.

8. Tuning performed at 120km/h:

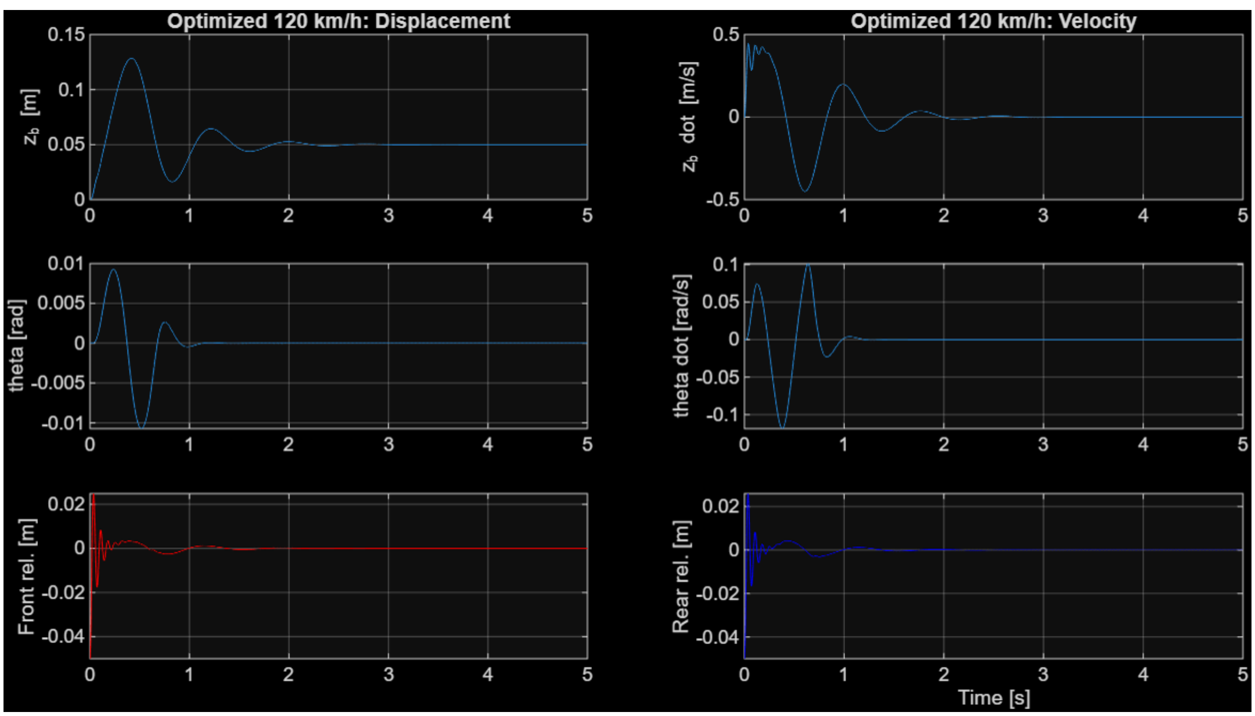

Fig 8 : `Half-Car optimizations at 120km/h`

Ride comfort at 120km/h demanded that we do some tuning to achieve a good ride of the half-car of 4-DOF model. I altered the values of suspension and damping to ensure that the chassis remained stable, prevented body oscillations, and provided sufficient compliance to the axles. I considered the front and rear suspension stiffness (k_{fk} and k_{rk}), damping coefficients (c_{fc} and c_{rc}), and tyre stiffness (k_{tfk} and k_{trk}) as the main variables.

The tuning loop was a sequence of sims at the desired speed. I considered the displacement and the velocity of the sprung load and unsprung load, and increased and decreased the suspension and damper values in turn. The idea was to achieve a flowing vertical movement of the car body with a low level of axle oscillations not to create poor ride comfort or difficult tyre-road contact.

Once the parameters were updated, the car body displacement and pitch were maintained at reasonable limits after a few repetitions. Short-lived oscillations quickly died and the velocities of unsprung masses remained moderate and this implied that road bumps were not so effective. What was left was a package that performs well at high speed, is comfortable, stable and manages well at 120km/h.

9. Validation and Practical Application

The model can be validated by comparing simulated displacements, velocities, and pitch angles with experimental data from instrumented test vehicles traversing bumps or standard road profiles. Accelerometer and suspension deflection measurements provide direct comparison. In practice, the results inform:

* Suspension design and optimization for target speeds.

* Ride comfort and handling assessment.

* Safety evaluation under transient road disturbances.

The 4-DOF half-car simulation effectively captures the essential dynamics of heave, pitch, and axle motion. It demonstrates that appropriate tuning of suspension and damping can optimize ride comfort and stability, even at high speeds. The technique provides engineers and directors with actionable insights for design decisions, vehicle testing, and performance optimization.

10. Summary :

This study presents a 4-DOF half-car model simulating the vertical and pitch dynamics of a car body and its unsprung masses over a road bump. Displacement and velocity responses were computed using forward Euler numerical integration with a 1 ms time step, ensuring convergence and numerical stability. Key results indicate that the sprung mass experiences smooth oscillations while the pitch angle captures realistic nose-up and nose-down rotations. Unsprung masses show higher-frequency responses due to tyre compliance. At a speed of 120 km/h, suspension and damping parameters were tuned to minimize body oscillations, limit axle velocity fluctuations, and ensure rapid decay of transients, achieving optimal ride comfort and vehicle stability.

References:

Gillespie, T. D. (1992). *Fundamentals of vehicle dynamics*. SAE International.

Rill, G. (2020). *Road vehicle dynamics: Fundamentals and modeling*. CRC Press.

Sun, L., & Chen, Z. (2007). Nonlinear vibration analysis of a half-car model. *Journal of Sound and Vibration, 302*(1), 69–82.

Peters, J., & Venhovens, P. J. (1999). Analysis of vehicle pitch dynamics using a half-car model. *Vehicle System Dynamics, 32*(1), 1–16.

Hrovat, D. (1997). Survey of advanced suspension developments and related optimal control applications. *Automatica, 33*(10), 1781–1817.

Appendix A: MATLAB Simulation

```
%% half_car_master.m

% Modular half-car master script

% Sections: eigenvalue analysis, base sim, speed sweep, q1..q4 plotting

clc; clear; close all;

%% ====================== 1. FLAGS (toggle sections) ======================

run_eigen       = true;   % eig analysis of linearised 4-DOF model
```

```
run_baseSim     = true;   % single simulation at base speed (50 km/h)
run_speedSweep  = true;   % multi-speed comparison (overlay)
run_qVariables  = true;   % q1..q4 formatted plots
%% ====================== 2. COMMON PARAMETERS ============================
% Mass / inertia
M   = 600;          % sprung mass (half-car) [kg]
J   = 300;          % pitch inertia [kg*m^2]
mf  = 40;           % front unsprung mass [kg]
mr  = 40;           % rear unsprung mass [kg]
% Stiffness & damping (tuned / as requested)
kf  = 2.0e4;        % front suspension stiffness [N/m]
kr  = 2.0e4;        % rear suspension stiffness [N/m]
cf  = 1000;         % front damping [N*s/m]
cr  = 1000;         % rear damping [N*s/m]
ktf = 3.0e5;        % front tyre stiffness [N/m]
ktr = 3.0e5;        % rear tyre stiffness [N/m]
% Geometry & gravity
lf  = 1.3;          % CoM to front axle [m]
lr  = 1.3;          % CoM to rear axle [m]
g   = 9.81;         % gravity [m/s^2]
% Road bump (user confirmed)
D0  = 0;            % start of bump [m]
W   = 20;           % bump width [m]
H   = 1;            % bump height [m]  <-- user choice
% Time discretization
dt    = 0.001;      % time step [s]
```

```
t_end = 5;        % simulation duration [s]

t     = 0:dt:t_end;

N     = length(t);

% Base speed (km/h -> m/s)

v_base_kmph = 50;

v_base = v_base_kmph / 3.6;

% Speeds for sweep (km/h)

speeds_kmph = [50, 100, 120, 160];

%% ====================== 3. EIGENVALUE ANALYSIS ==========================

if run_eigen

   fprintf('\n--- Eigenvalue analysis (4-DOF linearised) ---\n');

   % Map parameters into the simplified linear stiffness mapping used earlier

   K = (kf + kr)/2;        % representative suspension stiffness (assume
symmetric)

   Kt = (ktf + ktr)/2;     % representative tyre stiffness

   L_half = lf;            % mapping used previously (lf=lr=1.3)


   % Mass matrix (4 DOF: [zb, theta, zaxf, zaxr] mapping to [M, J, m, m])

   M_mat = [M     0     0    0;

            0     J     0    0;

            0     0     mf   0;

            0     0     0    mr];


   % Stiffness mapping consistent with earlier snippet

   % (the mapping uses K and geometric coupling via L_half)

   K_mat = [2*K       2*K*L_half   -K        -K;
```

```
        2*K*L_half 2*K*L_half^2 -K*L_half -K*L_half;
        -K       -K*L_half    K + Kt    0;
        -K       -K*L_half    0         K + Kt];

% Solve generalised eigenvalue problem
[phi, lambda] = eig(K_mat, M_mat);
lambda_vec = diag(lambda);

% Remove (or keep) tiny/negative numerical noise
lambda_vec(lambda_vec < 0) = 0;

omega = sqrt(lambda_vec);       % rad/s
freq_hz = omega / (2*pi);       % Hz

disp('Natural Frequencies (Hz):');
disp(freq_hz);
disp('Mode shapes (columns correspond to eigenvectors):');
disp(phi);

% Quick plot of mode participation (scaled)
figure('Name','Eigen Modes','NumberTitle','off');
for k = 1:size(phi,2)
    subplot(2,2,k);
    bar(abs(phi(:,k)));
    xticklabels({'zb','\theta','z_{axf}','z_{axr}'});
    ylabel('|mode amplitude|');
```

```
        title(sprintf('Mode %d: f = %.3f Hz', k, freq_hz(k)));

        grid on;

    end

end

%% ====================== 4. BASE SIMULATION (single speed) ==============

if run_baseSim

    fprintf('\n--- Base simulation at %d km/h ---\n', v_base_kmph);


    % Initialize state vector x = [zb; zb_dot; theta; theta_dot; zf; zf_dot;
zr; zr_dot]

    x = zeros(8, N);


    % Road inputs for base speed

    x_vehicle = v_base * t;

    zrf = roadInput(x_vehicle, D0, W, H);

    zrr = roadInput(x_vehicle - (lf + lr), D0, W, H);


    % Time stepping (forward Euler)

    for i = 1:N-1

        dx = halfcarEoM(x(:,i), M, J, mf, mr, kf, kr, cf, cr, ktf, ktr, lf, lr,
zrf(i), zrr(i));

        x(:,i+1) = x(:,i) + dx * dt;

    end


    % Extract signals for plotting

    zb   = x(1,:);    theta = x(3,:);

    zaxf = x(5,:);    zaxr = x(7,:);
```

```
    rel_front = zaxf - zrf;

    rel_rear  = zaxr - zrr;


    % Plot results (compact)

    figure('Name','Base Simulation (single speed)','NumberTitle','off');

    subplot(4,1,1);

    plot(t, zrf, 'r--', t, zrr, 'b--'); hold on;

    ylabel('Road elevation [m]');

    legend('Front road input','Rear road input');

    title(sprintf('Road Inputs at %d km/h (H=%.2f m)', v_base_kmph, H));

    grid on;


    subplot(4,1,2);

    plot(t, zb); ylabel('Body heave z_b [m]'); grid on;


    subplot(4,1,3);

    plot(t, theta*180/pi); ylabel('Pitch \theta [deg]'); grid on;


    subplot(4,1,4);

    plot(t, rel_front, 'r', t, rel_rear, 'b'); ylabel('Suspension deflection
[m]');

    legend('Front deflection','Rear deflection');

    xlabel('Time [s]'); grid on;

end

%% ====================== 5. SPEED SWEEP (overlay comparisons) ============

if run_speedSweep
```

```
    fprintf('\n--- Speed sweep: %s km/h ---\n', num2str(speeds_kmph));

    % Prepare figure

    figure('Name','Speed sweep
(overlay)','NumberTitle','off','Units','normalized','Position',[0.1 0.1 0.7
0.7]);


    % We'll overlay body heave and front deflection as examples

    subplot(2,1,1); hold on; grid on; title('Body heave: speed comparison');
ylabel('z_b [m]');

    subplot(2,1,2); hold on; grid on; title('Front suspension deflection: speed
comparison'); ylabel('Front deflection [m]');


    legends_heave = cell(1,length(speeds_kmph));

    legends_front = cell(1,length(speeds_kmph));

    for idx = 1:length(speeds_kmph)

        vk = speeds_kmph(idx);

        v_ms = vk/3.6;


        % simulate

        x = zeros(8,N);

        x_vehicle = v_ms * t;

        zrf = roadInput(x_vehicle, D0, W, H);

        zrr = roadInput(x_vehicle - (lf + lr), D0, W, H);

        for i = 1:N-1

            dx = halfcarEoM(x(:,i), M, J, mf, mr, kf, kr, cf, cr, ktf, ktr, lf,
lr, zrf(i), zrr(i));

            x(:,i+1) = x(:,i) + dx * dt;

        end
```

```
        zb = x(1,:);
        zaxf = x(5,:);
        rel_front = zaxf - zrf;

        % plot
        subplot(2,1,1);
        plot(t, zb);
        subplot(2,1,2);
        plot(t, rel_front);

        legends_heave{idx} = sprintf('v = %d km/h', vk);
        legends_front{idx} = sprintf('v = %d km/h', vk);
    end

    subplot(2,1,1);
    legend(legends_heave,'Location','best'); xlabel('Time [s]');
    subplot(2,1,2);
    legend(legends_front,'Location','best'); xlabel('Time [s]');
end
%% ====================== 6. q1..q4 formatted visualization ==============
if run_qVariables
    fprintf('\n--- q1..q4 formatted run (base speed) ---\n');
    % run at base speed
    x = zeros(8,N);
    x_vehicle = v_base * t;
```

```
   zrf = roadInput(x_vehicle, D0, W, H);

   zrr = roadInput(x_vehicle - (lf + lr), D0, W, H);

   for i = 1:N-1

       dx = halfcarEoM(x(:,i), M, J, mf, mr, kf, kr, cf, cr, ktf, ktr, lf, lr,
zrf(i), zrr(i));

       x(:,i+1) = x(:,i) + dx * dt;

   end


   % Map q variables

   q1 = x(1,:); q1d = x(2,:);

   q2 = x(3,:); q2d = x(4,:);

   q3 = x(5,:); q3d = x(6,:);

   q4 = x(7,:); q4d = x(8,:);

   rel_front = q3 - zrf; rel_rear = q4 - zrr;


   % Plot grid layout 4x2

   figure('Name','q1..q4
layout','NumberTitle','off','Units','normalized','Position',[0.05 0.05 0.85
0.85]);

   sgtitle('Half-Car: q1..q4 displacements and velocities (base speed)');

   subplot(4,2,1); plot(t, q1, 'b'); ylabel('q1 (zb) [m]'); grid on;

   subplot(4,2,2); plot(t, q1d, 'b'); ylabel('q1d [m/s]'); grid on;

   subplot(4,2,3); plot(t, q2*180/pi, 'r'); ylabel('q2 (\theta) [deg]'); grid
on;

   subplot(4,2,4); plot(t, q2d*180/pi, 'r'); ylabel('\theta_{dot} [deg/s]');
grid on;

   subplot(4,2,5); plot(t, q3, 'y'); ylabel('q3 (z_{axf}) [m]'); grid on;

   subplot(4,2,6); plot(t, q3d, 'y'); ylabel('q3d [m/s]'); grid on;
```

```
    subplot(4,2,7); plot(t, q4, 'm'); ylabel('q4 (z_{axr}) [m]'); grid on;

    subplot(4,2,8); plot(t, q4d, 'm'); ylabel('q4d [m/s]'); xlabel('Time [s]');
grid on;


    % Optional figure for road & relative defs

    figure('Name','Road and relative deflections','NumberTitle','off');

    subplot(3,1,1);

    plot(t, zrf, 'r--', t, zrr, 'b--'); ylabel('Road input [m]'); legend('Front
input','Rear input'); grid on;

    subplot(3,1,2); plot(t, rel_front); ylabel('Front suspension deflection
[m]'); grid on;

    subplot(3,1,3); plot(t, rel_rear); ylabel('Rear suspension deflection
[m]'); xlabel('Time [s]'); grid on;

end

%% ====================== 7. HELPER FUNCTIONS =============================

function dx = halfcarEoM(x, M, J, mf, mr, kf, kr, cf, cr, ktf, ktr, lf, lr,
zrf, zrr)

    % State vector: x = [zb; zb_dot; th; th_dot; zf; zf_dot; zr; zr_dot]

    zb = x(1); zb_dot = x(2);

    th = x(3); th_dot = x(4);

    zf = x(5); zf_dot = x(6);

    zr = x(7); zr_dot = x(8);

    % Relative motions

    delta_f = (zb + lf*th - zf);

    delta_r = (zb - lr*th - zr);

    delta_f_dot = (zb_dot + lf*th_dot - zf_dot);

    delta_r_dot = (zb_dot - lr*th_dot - zr_dot);

    % Forces (spring + damper)
```

```
    Fs_f = kf*delta_f + cf*delta_f_dot;

    Fs_r = kr*delta_r + cr*delta_r_dot;

    % Tyre forces (no tensile tyre force / simple linear)

    Ft_f = ktf * (zf - zrf);

    Ft_r = ktr * (zr - zrr);

    % Equations of Motion

    zb_ddot = (-Fs_f - Fs_r) / M;

    th_ddot = (-lf*Fs_f + lr*Fs_r) / J;

    zf_ddot = (Fs_f - Ft_f) / mf;

    zr_ddot = (Fs_r - Ft_r) / mr;

    dx = [zb_dot; zb_ddot; th_dot; th_ddot; zf_dot; zf_ddot; zr_dot; zr_ddot];

end

function z = STEP(d, d1, z1, d2, z2)

    % Smooth cubic step function between (d1,z1) and (d2,z2)

    if d <= d1

        z = z1;

    elseif d >= d2

        z = z2;

    else

        alpha = (d - d1) / (d2 - d1);

        z = z1 + (z2 - z1) * (alpha^2 * (3 - 2*alpha));

    end

end

function q = roadInput(d, D0, W, H)

    % Generates smooth bump: rises then falls using two STEP segments

    q = zeros(size(d));
```

```
   for ii = 1:length(d)

       q(ii) = STEP(d(ii), D0, 0, D0 + W/2, H) + STEP(d(ii), D0 + W/2, H, D0 +
W, 0);

   end

end
```