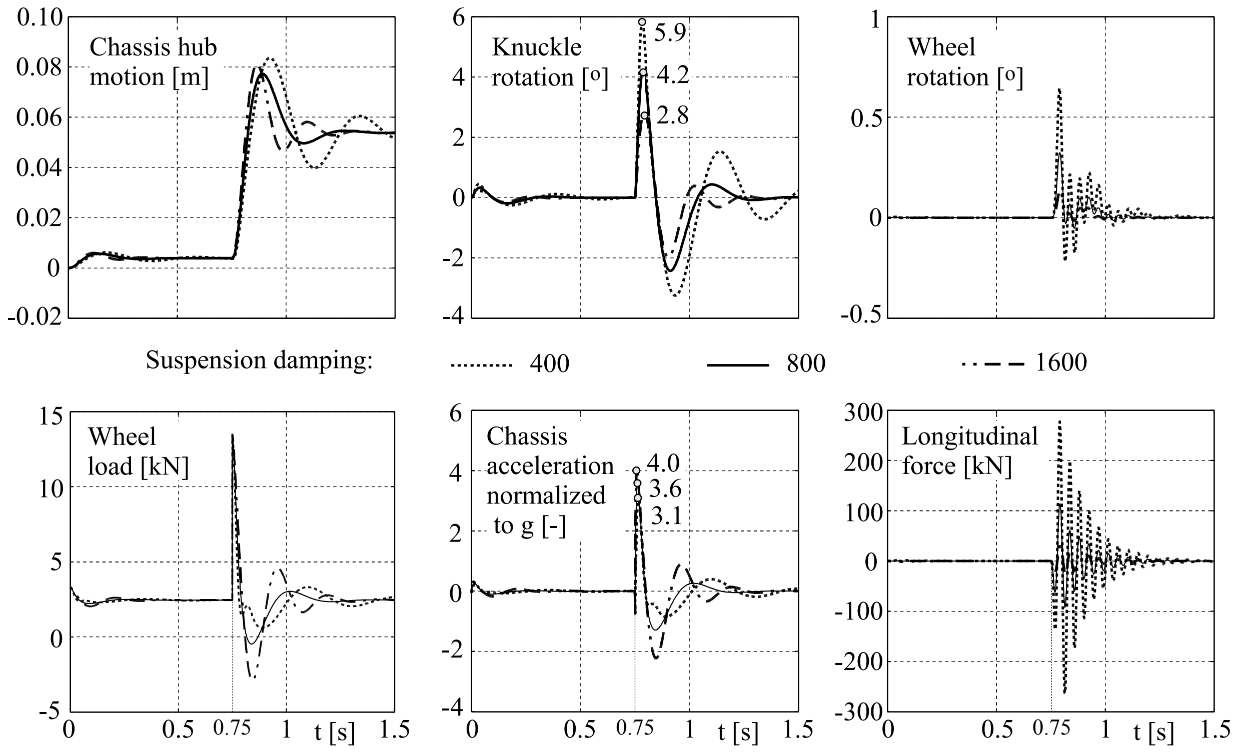

Figure S.2: Influence of damping

S.2 Exercises and Solution of Chapter Road

S.2.1 Cosine-shaped Bump

Problem: Modify the MATLAB-Script (Listing 2.3) and the corresponding function (Listing 2.2) so that the displacements of the actuator correspond with a cosine-shaped bump. Use the function in Listing 2.1 or the relevant statements in it. Note: The vehicle data are defined in the script qcm_data given in Listing 1.2.

Perform simulations with different heights (including negative values) and lengths of the obstacle. Replace the histogram-plot by the chassis acceleration which, will be provided by the fourth component of the state derivatives xdot. Keep in mind that valid results presume positive wheel loads!

Solution: The MATLAB-Script given in Listing S.2 runs the quarter car model across a cosine shaped obstacle and plots the results.

Listing S.2: Script `qcm_obstacle_main.m`: Quarter car model crossing an obstacle

```
global g s a b h r0
global mC mK mW ThetaK ThetaW
global Ts0 cs ds cx cz dx
global v_vel o_x0 o_y0 o_type o_width o_height o_length
```

```
 5 global z_road fz_wheel
 6
 7 % vehicle data
 8 qcm_data
 9
10 % veh. vel. as well as position, type and shape of obstacle
11 v_vel = 30/3.6; % vehicle velocity [km/h --> m/s]
12 o_x0 = 6.0; o_y0=0; % obstacle position [m]
13 o_type = 2;    % cosine shaped obstacle [-]
14 o_width =10;   % obstacle width [m]
15 o_height= 0.04; % obstacle height [m]
16 o_length= 5;   % obstacle length [m]
17
18 % initial states
19 x0 = [ 0; 0; 0; 0; 0; 0];
20
21 % time simulation
22 t0=0; tE=2; [tout,xout] = ode45(@qcm_obstacle_f,[t0,tE],x0);
23
24 % get road height and wheel load
25 zr=zeros(size(tout)); fz=zeros(size(tout)); zcdd=zeros(size(tout));
26 for i=1:length(tout)
27   xp = qcm_obstacle_f(tout(i),xout(i,:)’); zr(i)=z_road; fz(i)=fz_wheel; zcdd(i)=xp(4);
28 end
29
30 % plot results
31 subplot(3,2,1), plot(tout,zr), grid on, xlabel(’time’), legend(’road: z_R ’)
32 subplot(3,2,2), plot(tout,xout(:,2)*180/pi), grid on, xlabel(’time’), legend(’\beta [deg]’)
33 subplot(3,2,3), plot(tout,xout(:,1)), grid on, xlabel(’time’), legend(’chassis: z(t)’)
34 subplot(3,2,4), plot(tout,xout(:,3)*180/pi), grid on, xlabel(’time’), legend(’\phi [deg]’)
35 subplot(3,2,5), plot(tout,fz/1000), grid on, xlabel(’time’), legend(’wheel load F_z [kN]’)
36 subplot(3,2,6), plot(tout,zcdd), grid on, xlabel(’t [s]’), legend(’chassis acc. [m/s^2]’)
```

The data for the quarter car model are set via the script qcm_data.m. It is provided by Listing 1.2 and includes here the computation of the pre-load in the torsional spring that keeps the knuckle in a steady-state horizontal position. The code lines 12 to 16 define the position, the type and the shape of the obstacle. Finally, the dynamics of the quarter car model is provided by the function qcm_obstacle_f which is defined in Listing S.3.

Listing S.3: Function `qcm_obstacle_f.m`: Dynamics of a quarter car crossing an obstacle

```
function xp = qcm_obstacle_f(t,x)
% quarter car model with trailing arm suspension crossing an obstacle

global g s a b h r0
global mC mK mW ThetaK ThetaW
global Ts0 cs ds cx cz dx
global v_vel o_x0 o_y0 o_type o_width o_height o_length
global z_road fz_wheel

% state variables
zC = x(1); betaK = x(2); phiW = x(3); zCd = x(4); betaKd = x(5); phiWd = x(6);

% obstacle, centered @ o_x0, o_y0
sx = v_vel*t; sy=0;
```

```
15 u = obstacle_f(sx-o_x0,sy-o_y0,o_type,o_width,o_height,o_length); z_road = u;
16
17 % torque in revolute joint
18 Ts = - ( Ts0 + cs*betaK + ds*betaKd );
19
20 % tire deflection (static tire radius)
21 rS = h + zC - b + a*sin(betaK) - u ;
22
23 % longitudinal tire force (adhesion assumed)
24 Fx = - cx *( a*(1-cos(betaK)) - rS*phiW ) - dx *( a*sin(betaK)*betaKd - rS*phiWd ) ;
25
26 % vertical tire force (contact assumed)
27 Fz = cz *( r0 - rS ); fz_wheel = Fz;
28
29 % mass matrix
30 Massma=[   mC+mK+mW    (s*mK+a*mW)*cos(betaK) 0 ; ...
31       (s*mK+a*mW)*cos(betaK) ThetaK+s^2*mK+a^2*mW 0 ; ...
32              0                 0        ThetaW ];
33
34 % vector of generalized forces and torques
35 qgen=[ Fz-(mC+mK+mW)*g+(s*mK+a*mW)*sin(betaK)*betaKd^2 ; ...
36      Ts-(s*mK+a*mW)*cos(betaK)*g+a*(Fx*sin(betaK)+Fz*cos(betaK)); ...
37       -rS*Fx ];
38
39 % state derivatives
40 xp = [ zCd; betaKd; phiWd; Massma\qgen ];
41
42 end
```

The function obstacle_f, which is defined by Listing 2.1, computes the vertical road height as a function of the horizontal position of the vehicle or the contact point respectively. Here, the vehicle moves in longitudinal direction only. That is why the lateral coordinate sy is simply set to 0 in line 15.
As can be seen from Figure S.3, the impact on the vehicle is more severe at the shorter obstacle. When crossing the obstacle with a length of $L = 2.5\,m$ the wheel load becomes negative during a short time period. In practice, the wheel would lift off here.

S.2.2 Pseudo-random Road

Problem: Use the MATLAB-Script qcm_main.m (Listing 2.3) together with the MATLAB-Script qcm_data.m given in Listing 1.2 and the corresponding function qcm_f.m (Listing 2.2) to study the influence of the vehicle velocity on the wheel load.

Perform simulations with different values of the road power spectral density. Again, keep in mind that valid results presume positive wheel loads!

Solution: The MATLAB-Script given in Listing S.4 performs the simulation for the quarter car model with a random road input and plots the histogram of the wheel load. Again, the script qcm_data.m which set the data and is provided by Listing 1.2, is extended by the calculation of the pre-load in the torsional spring which keeps the knuckle in a horizontal position

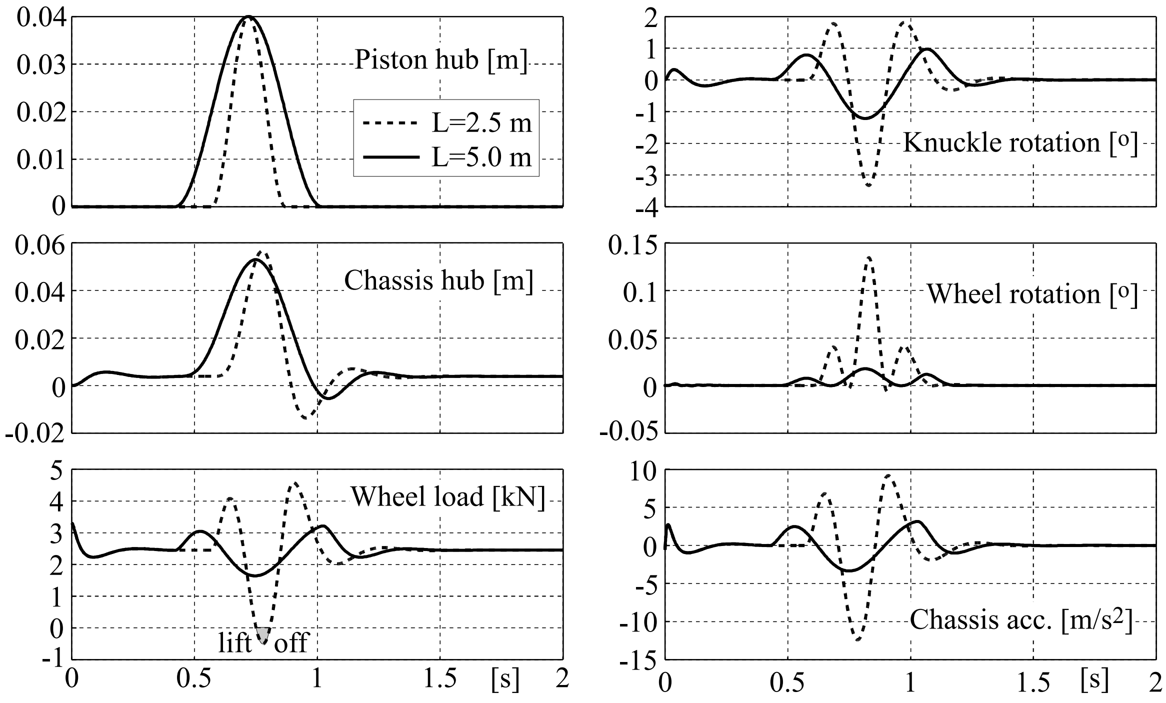

Figure S.3: Quarter car model crossing obstacles with different lengths

in steady state. The function qcm_f, which is given by Listing 2.2, computes the dynamics of the quarter car model and the road irregularities.

Listing S.4: Script `qcm_main.m`: Quarter car model on random road

```
global g s a b h r0
global mC mK mW ThetaK ThetaW
global Ts0 cs ds cx cz dx
global v_V Amp Om Psi
global z_road fz_wheel

% vehicle data
qcm_data

v_V = 90/3.6; % vehicle velocity [km/h --> m/s]
Phi0 = 2*10e-6; w = 2; % spectral density and waviness
Omin=0.0628; Omax=62.83; n=200; % range of frequencies and number of samples

% calculate amplitudes and random phases
dOm = (Omax-Omin)/(n-1); Om = Omin:dOm:Omax; Om0 = 1;
Phi = Phi0.*(Om./Om0).^(-w); Amp = sqrt(2*Phi*dOm); Psi = 2*pi*rand(size(Om));

% initial states: x0 = [ zC; betaK; phiW; zCd; betaKd; phiWd ]
x0 = [ 0; 0; 0; 0; 0; 0];

% simulation interval adjusted to vehicle velocity
t0=0; tE=150/v_V;
% adjust initial state to road
```

```
s_V = v_V*t0; u = sum( Amp.*sin(Om*s_V+Psi) ); x0(1)=u;
% perform time simulation
[tout,xout] = ode45(@qcm_f,[t0,tE],x0);

% get road height, wheel load and chassis acceleration
zr=zeros(size(tout)); fz=zeros(size(tout)); zdd=zeros(size(tout));
for i=1:length(tout)
  xdot = qcm_f(tout(i),xout(i,:)'); zr(i)= z_road; fz(i)=fz_wheel; zdd(i)=xdot(4);
end

fz_m=mean(fz); fz_std=std(fz); % mean value and standard deviation of wheel load
nh=20; [nfz,hfz]=hist(fz,nh); % generate histogram of wheel load

% generate gaussian density function and adjust to histogramm
fzi=linspace(min(fz),max(fz),201);
pfz = exp(-(fzi-fz_m).^2./(2*fz_std.^2))./(fz_std*sqrt(2*pi));
pfz = pfz * (sum(nfz)*(max(fz)-min(fz))/nh);

% plot only wheel load histogramm
barh(hfz/1000,nfz), colormap('white'), hold on, plot(pfz,fzi/1000,'k--','Linewidth',2)
title(['\sigma_F_z=',num2str(fz_std),', v=',num2str(v_V*3.6),', \Phi_0=',num2str(Phi0)])
```

The results for different vehicle velocities and road power spectral densities are shown in Figure S.4. In each simulation, the mean value of the wheel load m_{Fz} is close to the static wheel

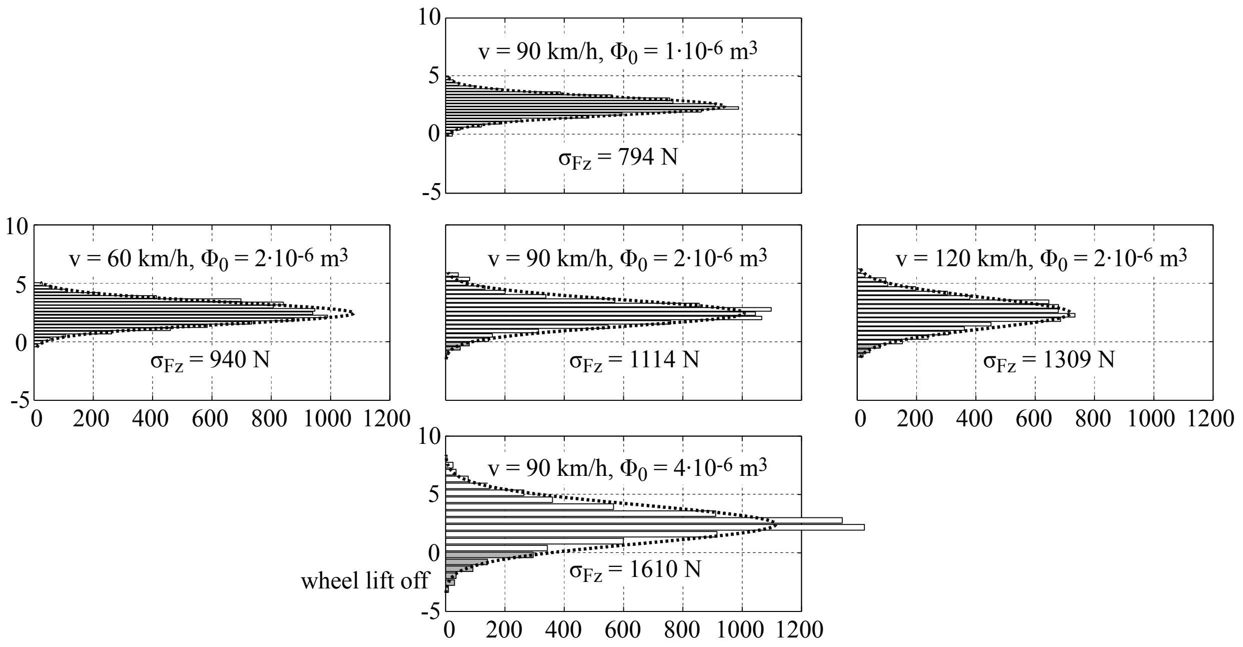

Figure S.4: Influence of vehicle velocity and road roughness on wheel load

load, which amounts to $F_z^{st} = 2.45\ kN$ here. It can be seen that the standard deviation σ_{Fz} of the wheel load increases with the vehicle velocity and the intensity of the road roughness. At $v = 120\ km/h$ and $\Phi_0 = 4 * 10^{-6}\ m^3$ a significant number of wheel lift offs occurs, which is indicated in the histogram by negative values of the wheel load. Here, the simulation results are not valid anymore because negative wheel loads are not possible in practice. Note that running the script qcm_main repeatedly will produce slightly different results because the MATLAB-Function rand generates each time a new set of random phase angles.

S.3 Exercises and Solution of Chapter Tire

S.3.1 Contact Geometry

Problem: The position of the wheel rotation axis with respect to the earth-fixed system is defined by the unit vector $e_{yR,0} = [0.097;\ 0.995;\ -0.024]^T$ in a particular driving situation. Calculate the unit vectors $e_{x,0}$ and $e_{y,0}$ pointing in the direction of the longitudinal and the lateral tire force as well as the tire camber angle γ when $e_{n,0} = [0;\ 0;\ 1]^T$ defines the track normal.

Solution: The basic approach to the contact geometry, described in section 3.2.1 of the text book provides in (3.6) and (3.7) the required equations. The code lines

```
eyR0 = [ 0.097; 0.995; -0.024 ]; % wheel rotation axis
en0  = [ 0.000; 0.000;  1.000 ]; % road normal
nx0 = cross(eyR0,en0); ex0=nx0/norm(nx0)    % longitudinal direction
ey0 = cross(en0,ex0) % lateral direction
ga = asin(eyR0'*en0); ga_grad=ga*180/pi  % camber angle [rad -> grad]
```

perform the corresponding computation in MATLAB and will result in

$$e_{x,0} = \begin{bmatrix} 0.9953 \\ -0.0970 \\ 0 \end{bmatrix}, \quad e_{y,0} = \begin{bmatrix} 0.0970 \\ 0.9953 \\ 0 \end{bmatrix}, \quad \gamma = -1.3752° \tag{S.13}$$

S.3.2 Contact Length

Problem: A tire with an unloaded radius of $r_0 = 0.546\,m$ is exposed to a vertical force of $F_z = 35\,kN$. Calculate the vertical tire stiffness c_z and the contact length L if a loaded or static tire radius of $r_S = 0.510\,m$ is measured.

Solution: The vertical tire stiffness is simply defined by

$$c_z = \frac{F_z}{\Delta_z} = \frac{F_z}{r_0 - r_S} = \frac{35\,kN}{0.546\,m - 0.510\,m} = 972.222\,kN/m = 972\,222\,N/m \tag{S.14}$$

and the contact length, which was estimated by Equations (3.42), is computed as

$$L = 2\sqrt{r_0 \Delta z} = 2\sqrt{r_0(r_0 - r_S)} = 2\sqrt{0.546\,m\,(0.546\,m - 0.510\,m)} = 0.28\,m \tag{S.15}$$

Wheel load and tire radius indicate a truck tire. That is why stiffness and contact length are comparatively large.