PROJECT 4
Kinematic behavior of a vehicle front suspension

TEAM MEMBERS

Barrasso Michele s270736

Bressani Riccardo s280878

Catel Nathalie Valois s306776

Colucci Carlo Vittorio s282350

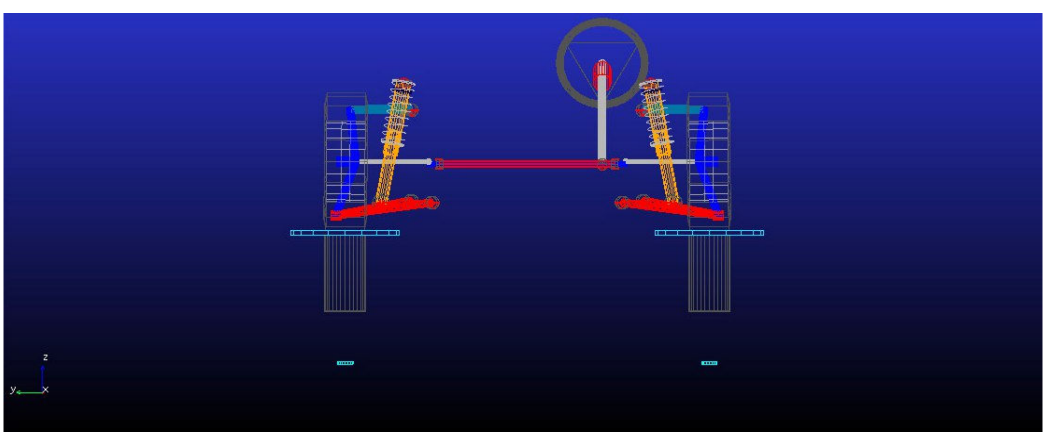
Covetti Alessio s281545

Placida Pierpaolo s281037

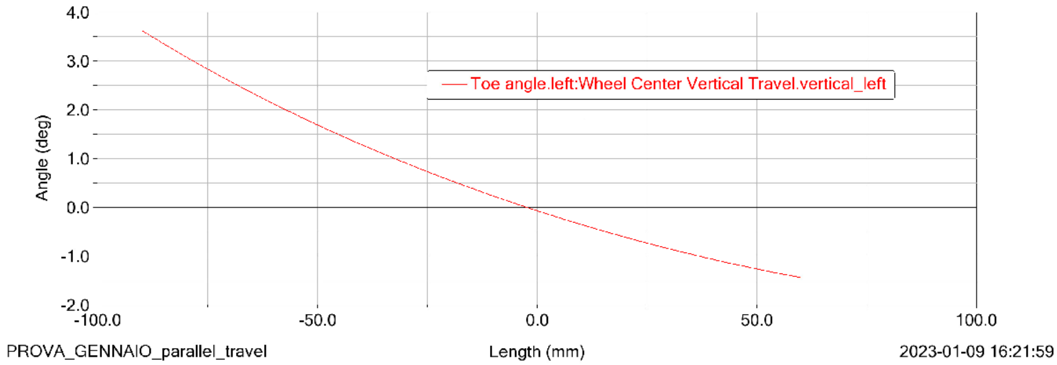
Vitale Michele s280970

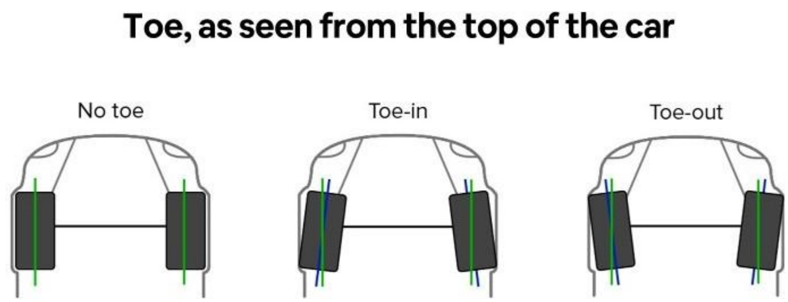
Abstract:
The aim of this project is to analyse the kinematic of a double whishbone suspension, through the aid of the Adams Car software and then to evaluate the forces acting on the spring. Moreover bumpstops are designed in order to obtain an increaseing stiffness when the suspension approches the limit regions of its travel. The forces and stifness at the ground are evaluated as well.

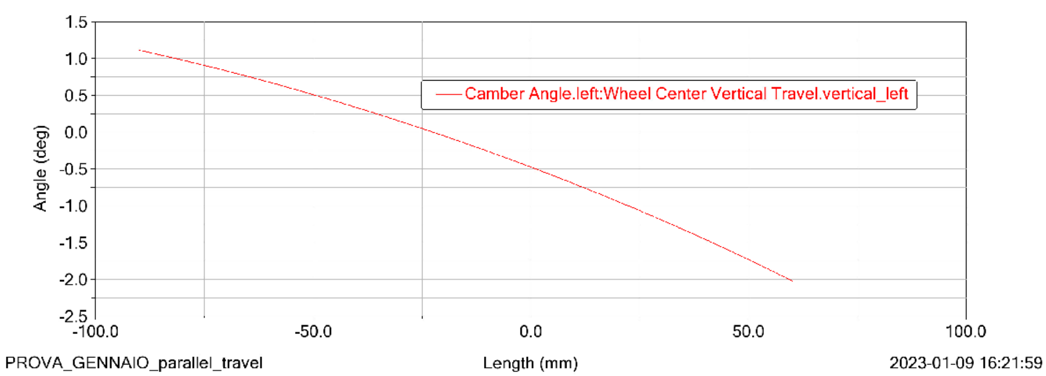
1) Kinematic behaviour of a front suspension:
From the geometrical data provided, it's possible to update the preset suspension model imported on Adams Car:

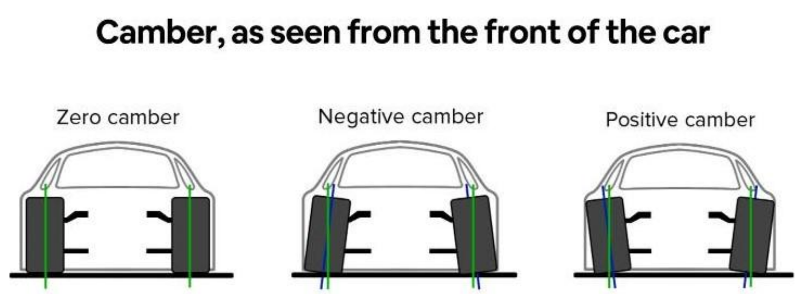


All the required simulations are run: first of all the changes in toe and camber angle and in semi-track, with varying wheel travel, are analysed:





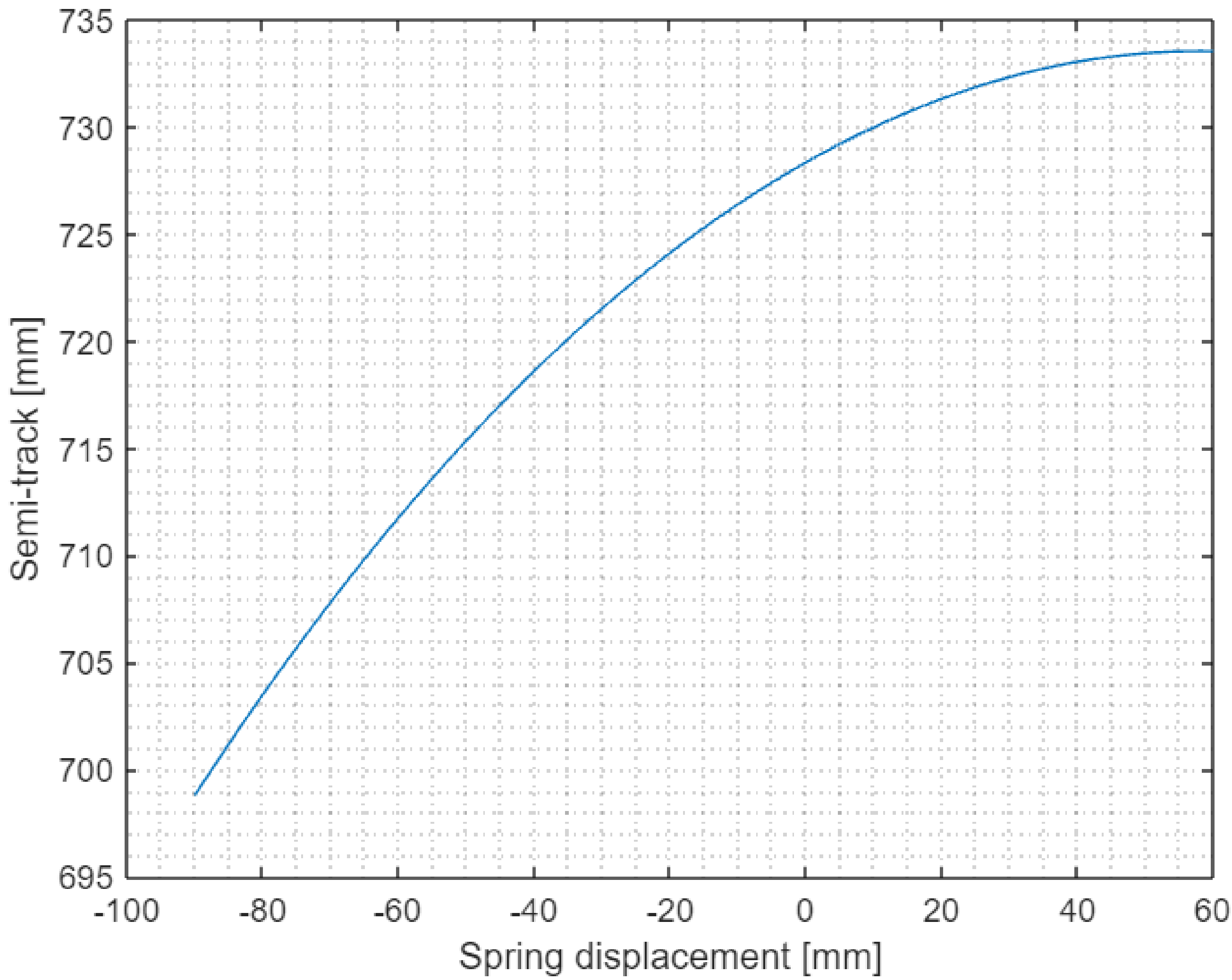




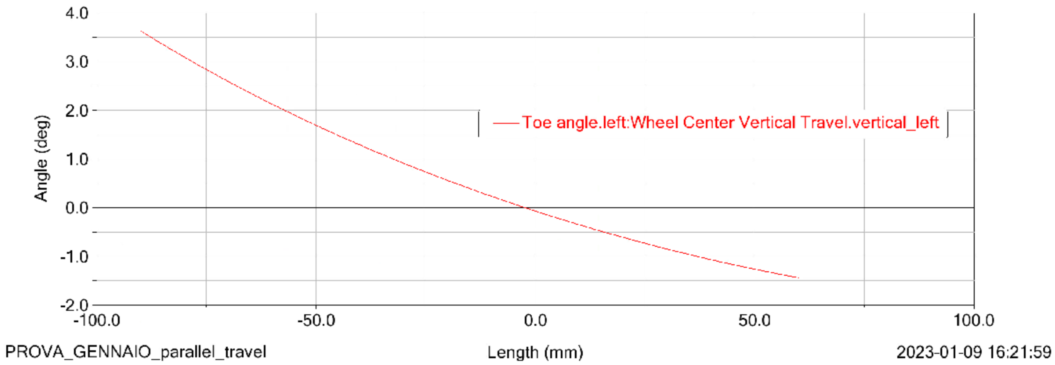
Adams Car's simulation returns the variation of the full track; to obtain the semi-track values, the data are imported on Matlab and divided by 2:

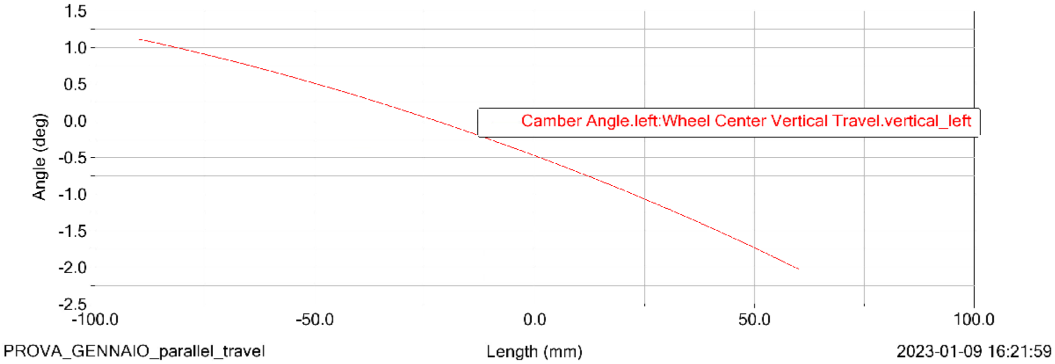
```
WheelTravel = readmatrix("Data analysis project 4.xlsx",'Sheet','Foglio1','Range','B2:B152');
Track = readmatrix("Data analysis project 4.xlsx",'Sheet','Foglio1','Range','D2:D152');

SemiTrack = Track / 2;
figure
plot(WheelTravel, SemiTrack);
xlabel('Spring displacement [mm]')
ylabel('Semi-track [mm]'); grid minor
```



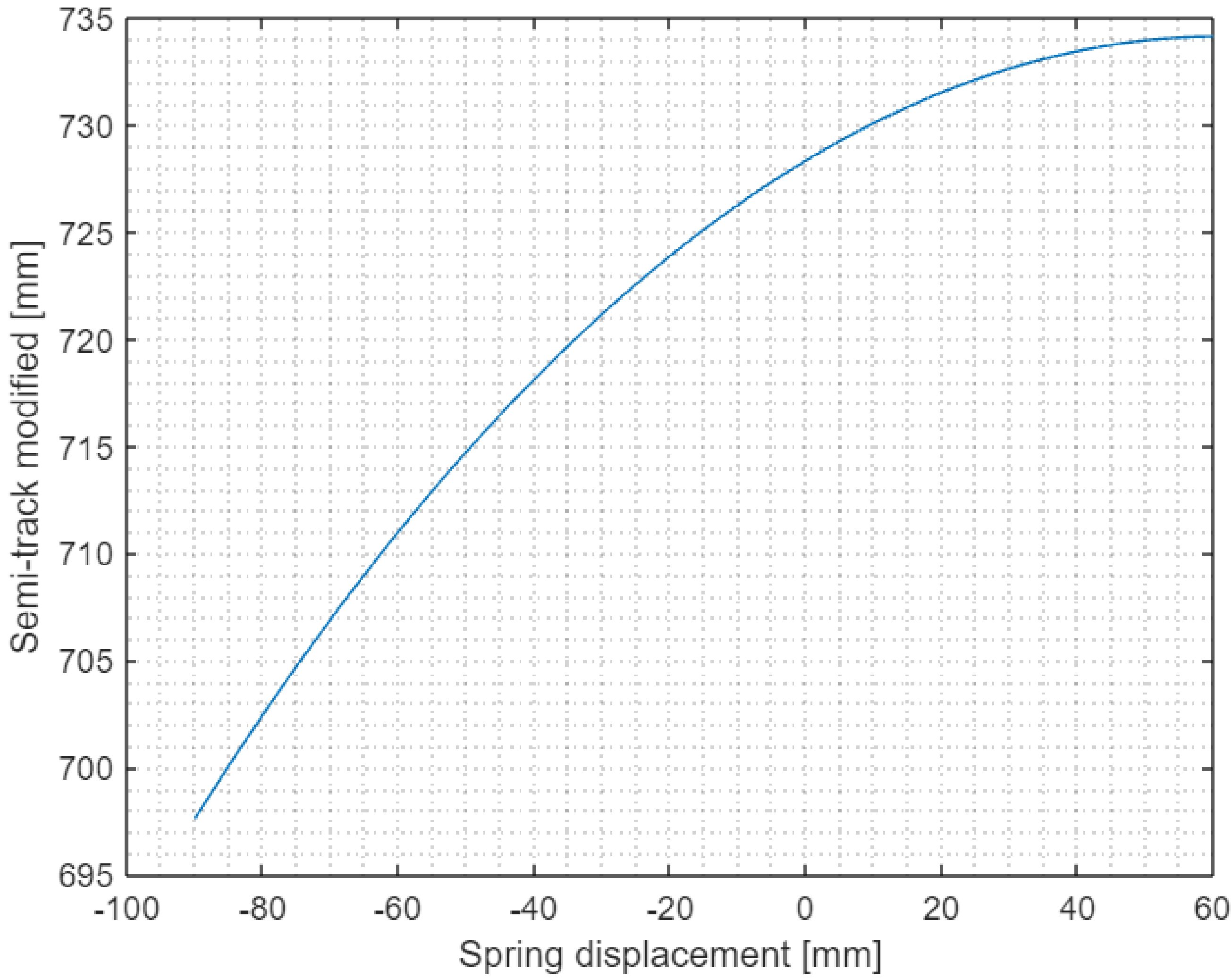
In order to evaluate the sensitivity of the suspension system, the height of the lower and upper control arms are changed (z(LCA)+10mm, z(UCA)-10mm), so new simulations are run and the updated plots are evaluated:






```
SpringDisplacement = readmatrix("Data analysis project 4.xlsx",'Sheet','Foglio1','Range','A2:A1
TrackModified = readmatrix("Data analysis project 4.xlsx",'Sheet','Foglio1','Range','E2:E152');

SemiTrackModified = TrackModified / 2;
figure
plot(WheelTravel, SemiTrackModified);
xlabel('Spring displacement [mm]')
ylabel('Semi-track modified [mm]'); grid minor
```



2) Static forces on the suspension:

The first step is to set the spring displacement to 0 when the wheel travel is none:

```
index = find(WheelTravel<0.1 & WheelTravel>-0.1); %tolerance imposed in order to find 0 value
DeltaSpringDisplacement = -(SpringDisplacement-SpringDisplacement(index)); % The minus sign is
```

The linear beahviour of the spring (no bump-stops) is defined in this way:

```
Ks = 24; % N/mm
Fs0 = 4195; % N
Fs = Fs0 + Ks*DeltaSpringDisplacement; % N
figure;
plot(DeltaSpringDisplacement, Fs)
title('Spring response without bump-stops')
xlabel('Spring displacement [mm]')
ylabel('Spring force [N]'); grid minor
xlim([-50, 40])
```

```
ylim([2000, 6000])
```

To design the bump-stops the followiong parameters are defined:

Spring displacement points:

```
MaxBound = DeltaSpringDisplacement(end); % last value of the vector
MinRebound = DeltaSpringDisplacement(1); % 1st value of the vector
MeanDisplacement = (MaxBound + MinRebound)/2;
LimBound = MaxBound - (MaxBound-MeanDisplacement)/3;
LimRebound = MinRebound - (MinRebound-MeanDisplacement)/3;
```

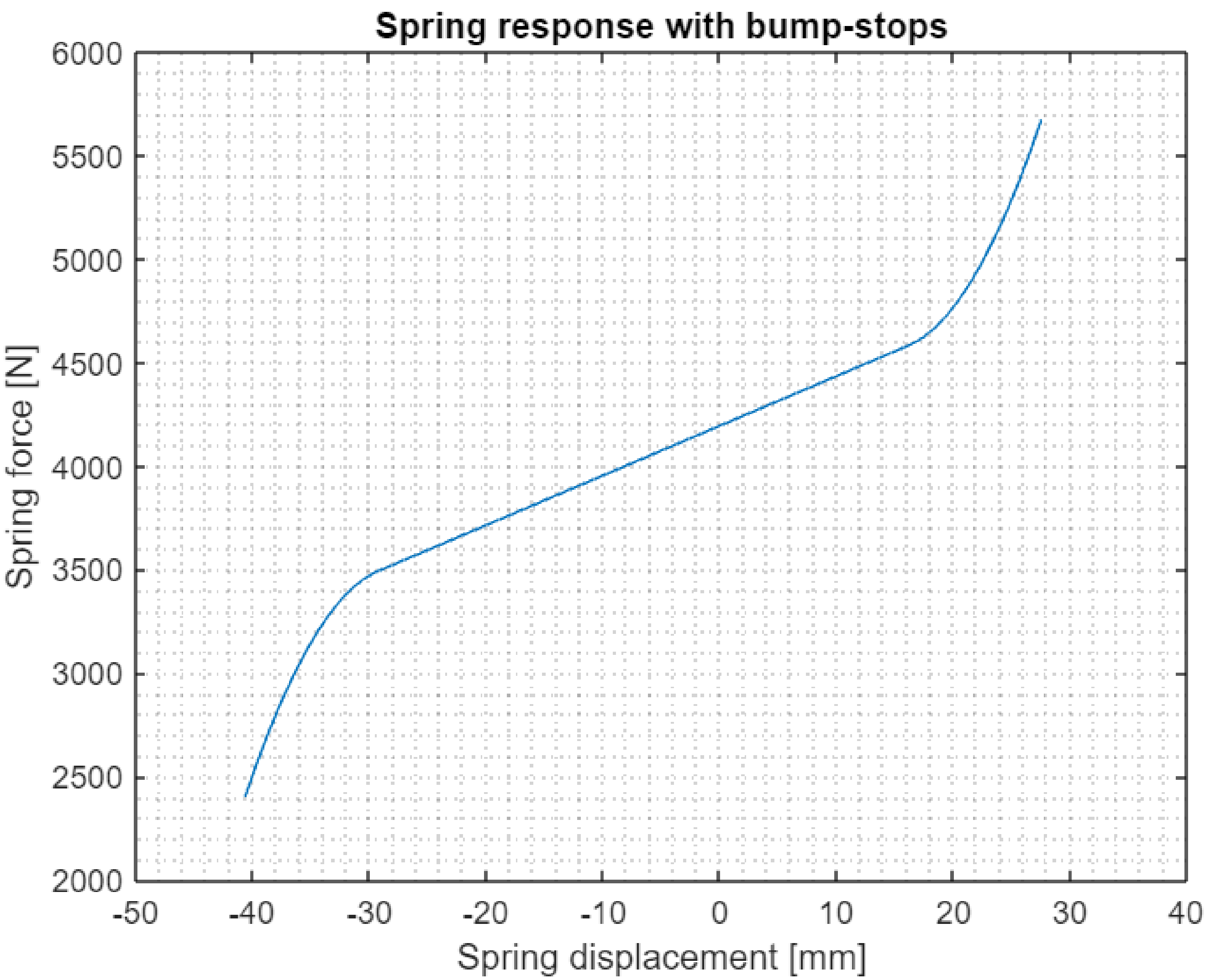
Computation of the bump-stops' coefficient:

```
Kb = Ks * (MaxBound-MeanDisplacement) / (MaxBound-LimBound)^2;
% In the data coming from the simulation, the closest to the ones calculated are searched for
IndexLimRebound = find(abs(DeltaSpringDisplacement-LimRebound)<0.2);
IndexLimBound = find(abs(DeltaSpringDisplacement-LimBound)<0.2);
% The vectors for the bump-stop regions are then defined
EndStopRebound = DeltaSpringDisplacement(1:IndexLimRebound);
```

```
EndStopBound = DeltaSpringDisplacement(IndexLimBound:end);
```

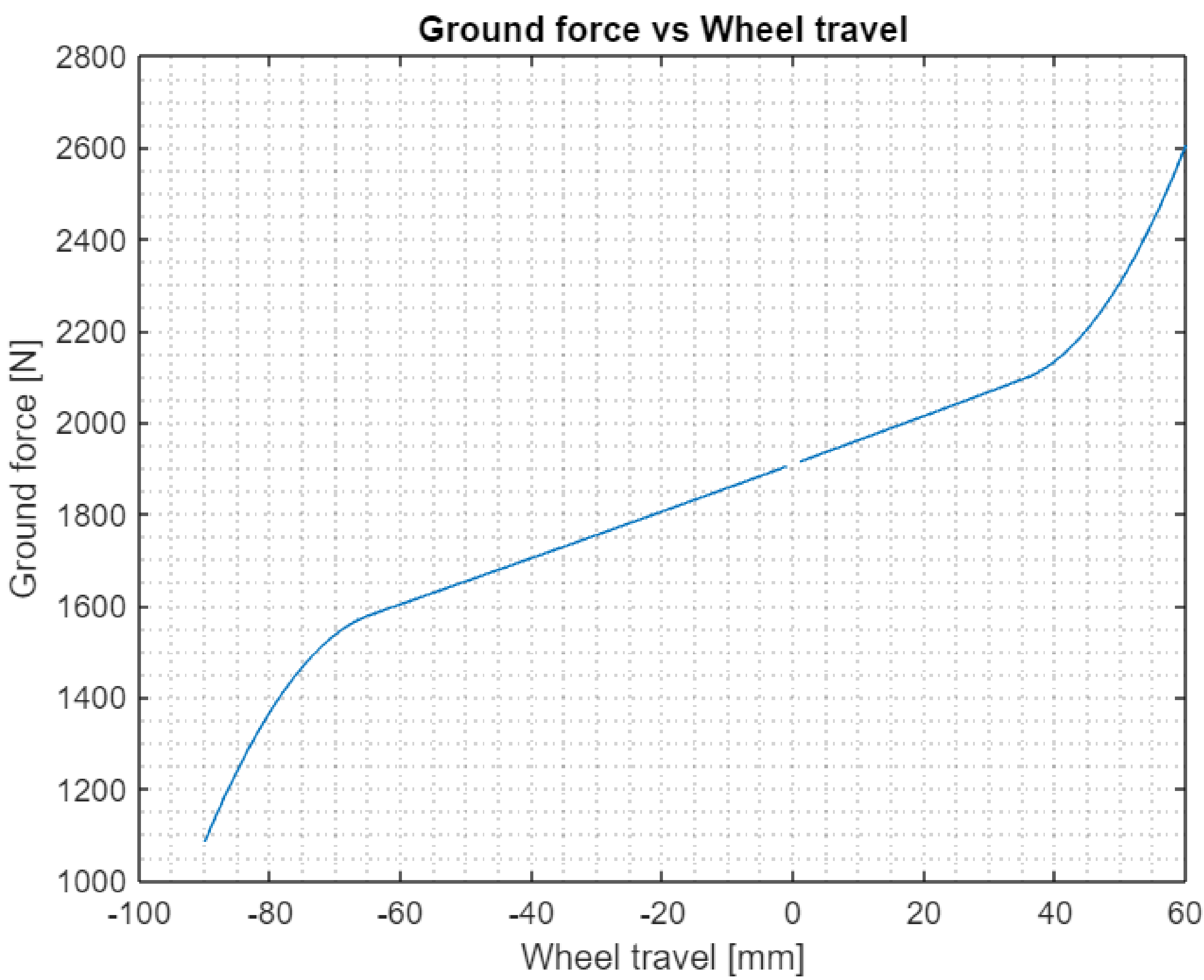
The total force is then computed; notice that in the bump-stop regions the latter is given by the sum of the linear contribution and the further quadratic term:

```
FbRebound = Fs(1:IndexLimRebound) - Kb * (EndStopRebound-LimRebound).^2;
FbBound = Fs(IndexLimBound:end) + Kb * (EndStopBound - LimBound).^2;
Ftot = [FbRebound; Fs(IndexLimRebound+1 : IndexLimBound-1); FbBound]; % unified vector of the t
figure
plot(DeltaSpringDisplacement, Ftot)
title('Spring response with bump-stops')
grid minor
xlim([-50, 40])
ylim([2000, 6000])
xlabel('Spring displacement [mm]')
ylabel('Spring force [N]');
```



The vertical force at the ground is now calculated and shown:

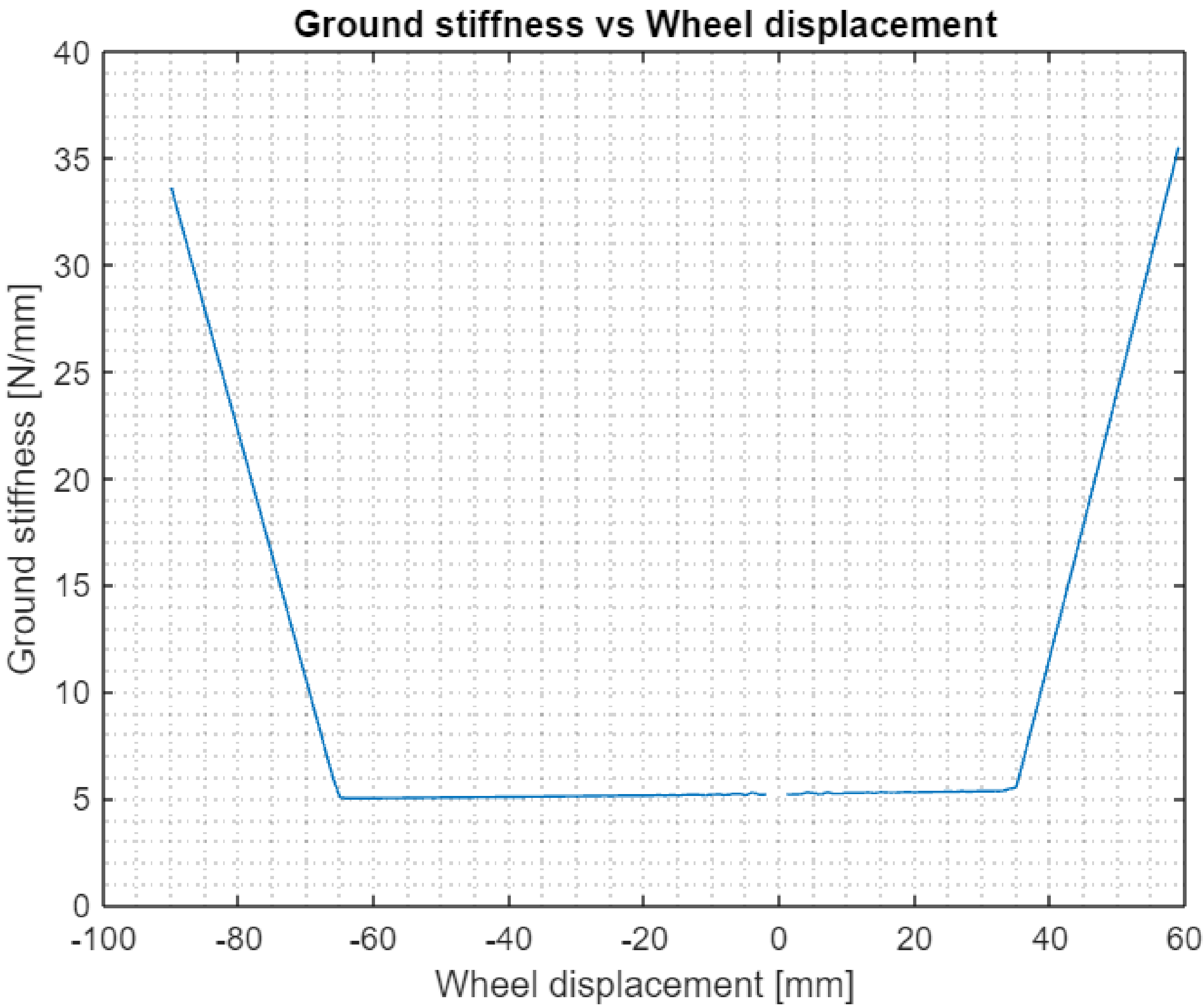
```
Fg = Ftot .* DeltaSpringDisplacement ./ WheelTravel; % from energy balance
Fg(91)=NaN; % this term corresponds to a division 0/0
figure
plot(WheelTravel, Fg); grid minor
title('Ground force vs Wheel travel')
xlabel('Wheel travel [mm]')
ylabel('Ground force [N]');
```



Finally, the ground stiffness is evaluated as $\frac{\delta F_z}{\delta z}$:

```
% Derivative of Fg with respect to z is computed analitically
Kg = [];
for i = 2:length(Fg)
    Kg = [Kg, (Fg(i)-Fg(i-1))/(WheelTravel(i)-WheelTravel(i-1))];
end
figure
plot(WheelTravel(1:end-1), Kg); grid minor
title('Ground stiffness vs Wheel displacement')
```

```
ylim([0,40])
xlabel('Wheel displacement [mm]')
ylabel('Ground stiffness [N/mm]');
```




```
ans = 1×16
  -90.0000  -80.0000  -70.0000  -60.0000  -50.0000  -40.0000  -30.0000  -20.0000 ⋯
```

The data returned by the simulation are now reported: