VIENNA UNIVERSITY OF TECHNOLOGY

FACULTY OF MECHANICAL AND INDUSTRIAL ENGINEERING

INSTITUTE OF MECHANICS AND MECHATRONICS

PROJECT WORK ENGINEERING DYNAMICS

REPORT

MICHAL ANDO

M.N.1327785

MARTIN SIVÝ

M.N.1327784

2013/2014 WS

Introduction

Aim of this work is to derive the equation of motion for vehicle with trailer. Derived equation of motion will be verified by comparison of the natural frequencies computed in Matlab with the natural frequencies computed in MSC.adams.

Other task is to simulate the transition trough the bump modelled as half sine. Results from simulation will bee also verified in MSC.adams.

Trailer is used for cargo transportation. Has two axels with rubber torsion springs and is shown on the figure 1.:

Figure 1. Trailer

Problem description:

1. Construction of the mathematical model and derivation of the equation of motion.
2. Computation of the natural frequencies and mode shapes in Matlab environment.
3. Construction of the alternative model in MSC.adams environment and computation of the natural frequencies and mode shapes.
4. Comparison of the natural frequencies and mode shapes results.
5. Road input definition and model response simulation in Matlab.
6. Road input definition and model response simulation in MSC.adams.
7. Comparison of the model response results.

Assumptions:

1. Vehicle and trailer are symmetric with respect to longitudinal axis therefore half vehicle trailer model will be investigated.
2. Model is linearized due to small angle assumptions.
3. Damping of the wheels is neglected.
4. Stiffness, damping, inertia parameters and the position of the centres of the gravity are not corresponding to the investigated vehicle and trailer but are chosen approximately.
5. Positions of the springs and position of the coupling between vehicle and trailer are measured.
6. Damping and stiffness characteristic of the vehicles dashpots and springs are linear.

1. Construction of the mathematical model and derivation of the equation of motion

To build the mathematical model three degrees of freedom were considered. One translational degree of freedom - vertical displacement of the vehicles centre of the gravity and two rotational degrees of freedom - rotation of the vehicles and trailers centres of the gravity.

Equation of motion were obtained directly from relations for kinetic and potential energy. Centre of the coordinate system was set to the vehicles centre of the gravity.

Coordinates of the vehicle and trailer centre of the gravity were calculated with respect to the coupling between the vehicle and trailer. Derivatives of this coordinates were used to compute the kinetic energy.

Potential energy was computed from the spring deflection.

Figure 2. represents the mathematical model of the vehicle with trailer:

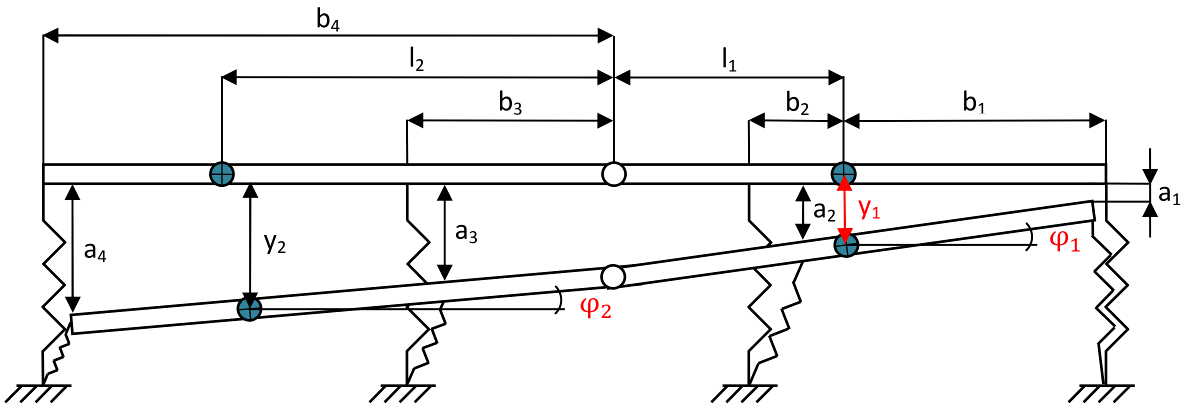

Figure 2. Half vehicle trailer model

Coordinates of the vehicle and trailer centre of the gravity:

$$x_1 = 0$$
$$y_1 = y_1(t)$$
$$x_2 = cos\varphi_1(t) \cdot l_1 + cos\varphi_2(t) \cdot l_2$$
$$y_2 = y_1(t) + sin\varphi_1(t) \cdot l_1 + sin\varphi_2(t) \cdot l_2$$

Derivatives of the coordinates of the vehicle and trailer centre of the gravity:

$$\dot{y}_1 = \dot{y}_1(t)$$
$$\dot{x}_2 = -sin\varphi_1(t) \cdot \dot{\varphi}_1(t) \cdot l_1 - sin\varphi_2(t) \cdot \dot{\varphi}_2(t) \cdot l_2$$
$$\dot{y}_2 = \dot{y}_1(t) + cos\varphi_1(t) \cdot \dot{\varphi}_1(t) \cdot l_1 + cos\varphi_2(t) \cdot \dot{\varphi}_2(t) \cdot l_2$$

Expression for the kinetic energy:

$$T = \frac{1}{2} \cdot m_1 \cdot \dot{y}_1^2 + \frac{1}{2} \cdot m_2 \cdot (\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2} \cdot I_1 \cdot \dot{\varphi}_1^2 + \frac{1}{2} \cdot I_2 \dot{\varphi}_2^2$$

Kinetic energy expression after algebraic operations and linearization:

$$T = \frac{1}{2}[m_1 \dot{y}_1^2 + m_2(\dot{y}_1^2 + \dot{\varphi}_1^2 l_1^2 + \dot{\varphi}_2^2 l_2^2 + 2\dot{\varphi}_1 \dot{\varphi}_2 l_1 l_2 + 2\dot{\varphi}_1 \dot{y}_1 l_1 + 2\dot{\varphi}_2 \dot{y}_1 l_2) + I_1 \dot{\varphi}_1^2 + I_2 \dot{\varphi}_2^2]$$

Coordinates of the vertical spring displacement:

$$a_1 = y_1(t) - sin\varphi_1(t) \cdot b_1$$
$$a_2 = y_1(t) + sin\varphi_1(t) \cdot b_2$$
$$a_3 = y_1(t) + sin\varphi_1(t) \cdot l_1 + sin\varphi_2(t) \cdot b_3$$
$$a_4 = y_1(t) + sin\varphi_1(t) \cdot l_1 + sin\varphi_2(t) \cdot b_4$$

Expression for the potential energy:

$$V = \frac{1}{2} \cdot k_1 \cdot a_1^2 + \frac{1}{2} \cdot k_2 \cdot a_2^2 + \frac{1}{2} \cdot k_3 \cdot a_3^2 + \frac{1}{2} \cdot k_4 \cdot a_4^2$$

Potential energy expression after algebraic operations and linearization:

$$V = \frac{1}{2} k_1 (y_1^2 - 2y_1\varphi_1 b_1 + \varphi_1^2 b_1^2) + \frac{1}{2} k_2 (y_1^2 + 2y_1\varphi_1 b_2 + \varphi_1^2 b_2^2) + \frac{1}{2} k_3 (y_1^2 + + 2y_1\varphi_1 l_1 + 2y_1\varphi_2 b_3 + \varphi_1^2 l_1^2 + 2\varphi_1\varphi_2 l_1 b_3 + \varphi_2^2 b_3^2) + \frac{1}{2} k_4 (y_1^2 + 2y_1\varphi_1 l_1 + 2y_1\varphi_2 b_4 + + \varphi_1^2 l_1^2 + 2\varphi_1\varphi_2 l_1 b_4 + \varphi_2^2 b_4^2)$$

Mass and stiffness matrix were obtained directly from the kinetic and potential energy by following formulas:

$$m_{ij} = \frac{\partial^2 T}{\partial \dot{q}_i \partial \dot{q}_j}$$
$$k_{ij} = \frac{\partial^2 V}{\partial q_i \partial q_j}$$

Resulting mass and stiffness matrixes are:

$$\boldsymbol{M} = \begin{bmatrix} m_1 + m_2 & m_2 l_1 & m_2 l_2 \\ m_2 l_1 & m_2 l_1^2 + I_1 & m_2 l_1 l_2 \\ m_2 l_2 & m_2 l_1 l_2 & m_2 l_2^2 + I_2 \end{bmatrix}$$
$$\boldsymbol{K} = \begin{bmatrix} k_1 + k_2 + k_3 + k_4 & -k_1 b_1 + k_2 b_2 + k_3 l_1 + k_4 l_1 & k_3 b_3 + k_4 b_4 \\ -k_1 b_1 + k_2 b_2 + k_3 l_1 + k_4 l_1 & k_1 b_1^2 + k_2 b_2^2 + k_3 l_1^2 + k_4 l_1^2 & k_3 l_1 b_3 + k_4 l_1 b_4 \\ k_3 b_3 + k_4 b_4 & k_3 l_1 b_3 + k_4 l_1 b_4 & k_3 b_3^2 + k_4 b_4^2 \end{bmatrix}$$

Corresponding equations of motion can be written in matrix form:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{0}, \qquad \ddot{\boldsymbol{q}} = \begin{bmatrix} \ddot{y}_1 \\ \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \end{bmatrix}, \qquad \boldsymbol{q} = \begin{bmatrix} y_1 \\ \varphi_1 \\ \varphi_2 \end{bmatrix}, \qquad \boldsymbol{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

2. Computation of the natural frequencies in Matlab environment

Mass and stiffness matrixes were constructed in Matlab. Command *eig(K,M)* was used to compute natural frequencies and mode shapes. Script is included in appendix where are also shown the values of used parameters

Natural frequencies computed in Matlab are:

$f_1 = 1.2874\ Hz$

$f_2 = 1.5179\ Hz$

$f_3 = 3.1989\ Hz$

3. Construction of the alternative model in MSC.adams environment and computation of the natural frequencies

MSC.adams is a simulation software for multi-body dynamics. Its capabilities are much higher than our problem requires. Goal is to make a model that corresponds as much as possible to the model from Matlab.

Our model consists of two parts that represent vehicle and trailer. Parts are connected by rotational joint that describes the coupling between vehicle and trailer. Four springs are connected to the parts and to the ground.

Dimensional parameters (position of the springs, coupling and centres of the gravity), stiffness parameters of the springs and the inertia parameters (masses and moments of inertia) are equal to parameters used in Matlab.

To the vehicles centre of the gravity is connected mass less part by rotational joint. This part is also connected by translational joint to the ground. This solution allows to the vehicle rotational and vertical translational degree of freedom and the third degree of freedom is the rotation of the trailer. This composition corresponds as much as possible to the model from Matlab.

Figure 3. shows the model build in MSC.adams.

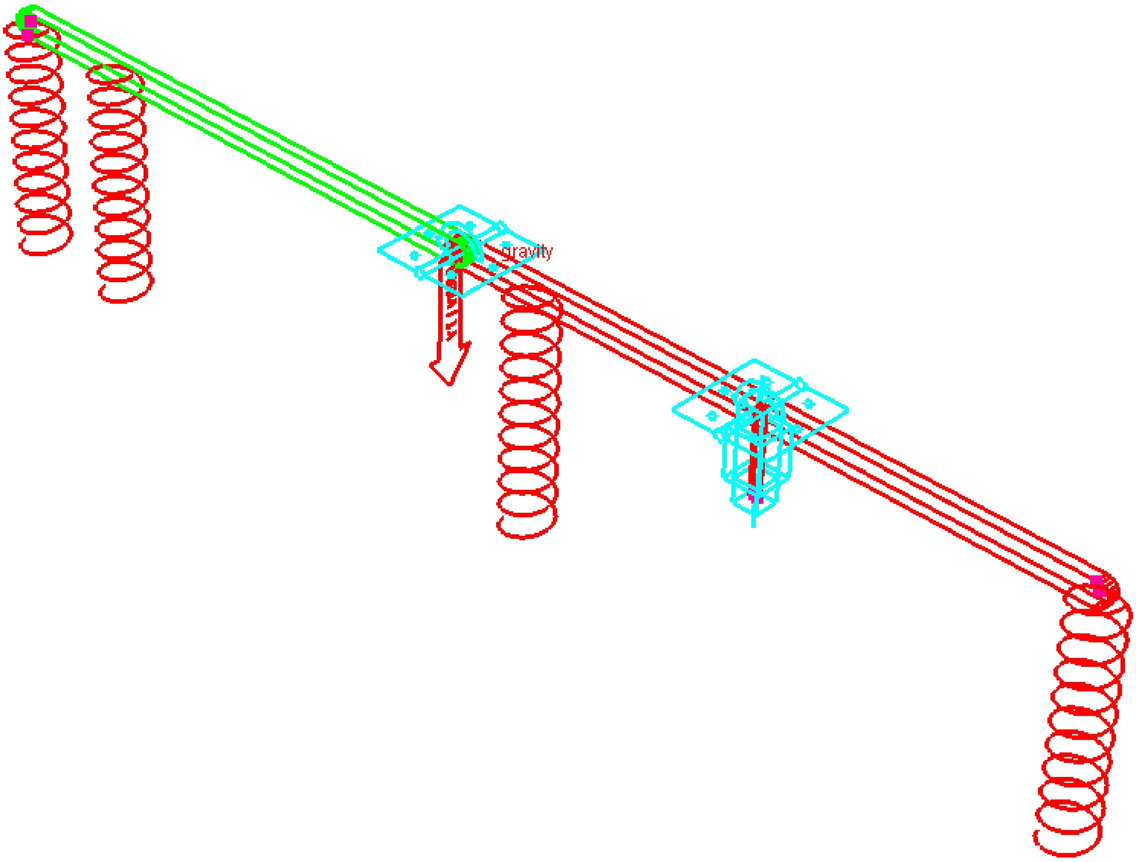

Figure 3. Model build in MSC.adams.

Natural frequencies computed in MSC.adams are:

$f_1 = 1.28624\ Hz,$

$f_2 = 1.51759\ Hz$

$f_3 = 3.19785\ Hz.$

4. Comparison of the natural frequencies and mode shapes results.

Absolute differences between natural frequencies computed in Matlab and in MSC.adams are:

$|\Delta f_1| = 0.0012\ Hz$

$|\Delta f_2| = 0.00027076\ Hz$

$|\Delta f_3| = 0.0011\ Hz$

This results are very satisfying and they indicates that equation of motion are correct. Comparison of the mode shapes is displayed on the figure 4.:

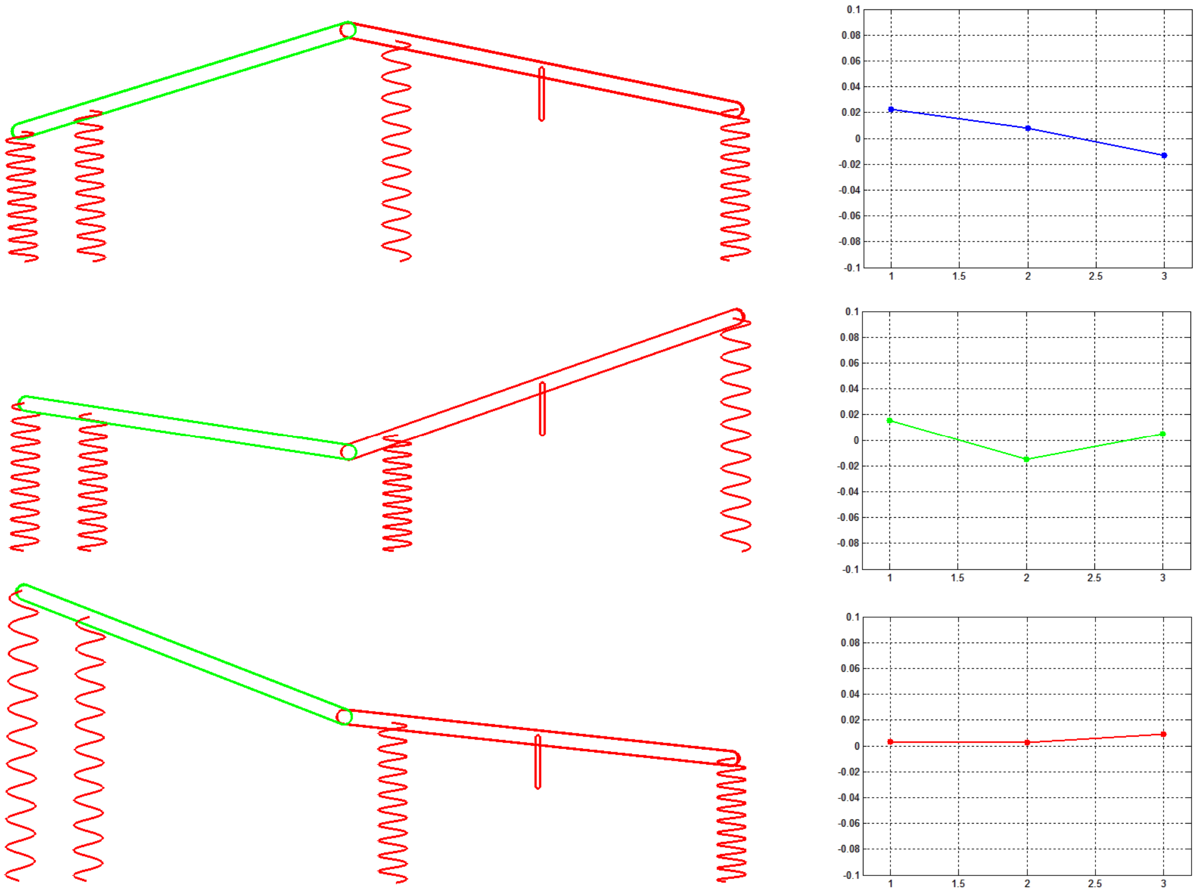

Figure 4. Mode shapes visualization left MSC.adams, right Matlab

5. Road input definition and model response simulation in Matlab.

5.1. Introduction of damping into the model

Damping matrix is equivalent with the stiffness matrix and can be written in the form:

$$\boldsymbol{C}=\begin{bmatrix} c_1+c_2+c_3+c_4 & -c_1b_1+c_2b_2+c_3l_1+c_4l_1 & c_3b_3+c_4b_4 \\ -c_1b_1+c_2b_2+c_3l_1+c_4l_1 & c_1b_1^2+c_2b_2^2+c_3l_1^2+c_4l_1^2 & c_3l_1b_3+c_4l_1b_4 \\ c_3b_3+c_4b_4 & c_3l_1b_3+c_4l_1b_4 & c_3b_3^2+c_4b_4^2 \end{bmatrix}$$

Our equation of motion including the damping matrix has a form:

$$\boldsymbol{M}\ddot{\boldsymbol{q}}+\boldsymbol{C}\dot{\boldsymbol{q}}+\boldsymbol{K}\boldsymbol{q}=\boldsymbol{0}, \qquad \ddot{\boldsymbol{q}}=\begin{bmatrix}\ddot{y}_1\\ \ddot{\varphi}_1\\ \ddot{\varphi}_2\end{bmatrix}, \qquad \dot{\boldsymbol{q}}=\begin{bmatrix}\dot{y}_1\\ \dot{\varphi}_1\\ \dot{\varphi}_2\end{bmatrix}, \qquad \boldsymbol{q}=\begin{bmatrix}y_1\\ \varphi_1\\ \varphi_2\end{bmatrix}, \qquad \boldsymbol{0}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

5.2. Definition of road profile and corresponding input functions

Our target is to simulate transition of the vehicle with trailer trough a bump. Bump is modelled as a half-sine with characteristic height and length. Bump is shown on the figure 4.:

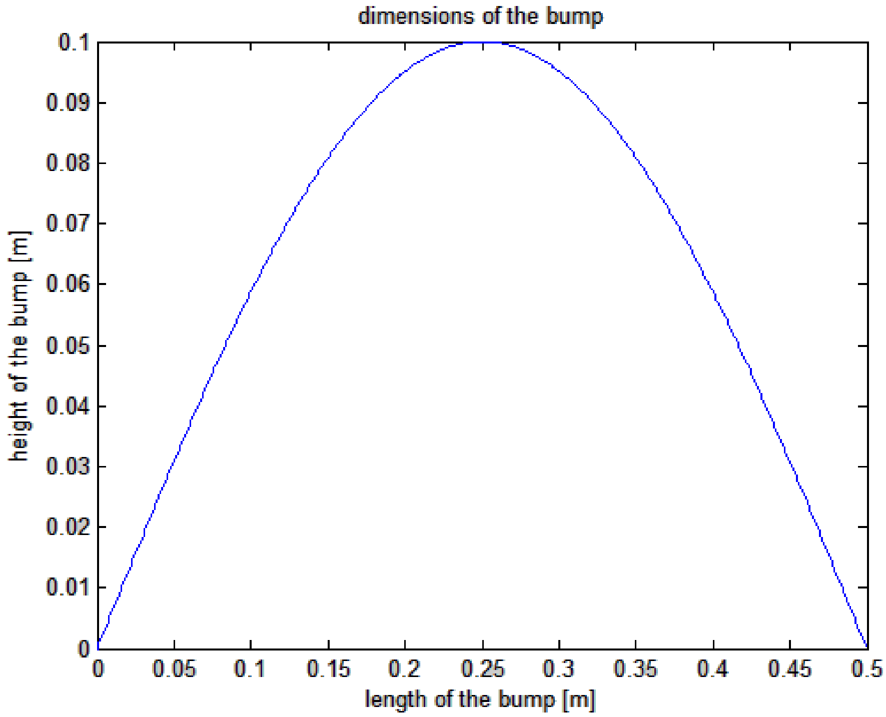

Figure 4. Dimensions of the bump

To model the transition of the vehicle with trailer trough the bump we assume constant velocity. Our modelling approach is based on four input function. Figure 5. represents times at approaching and leaving the bump for each wheel. Computed times are based on vehicles and trailers dimensions and on the velocity. Small *d* stands for the length of the bump.

	Time at approaching [s]	Time at leaving [s]
Vehicles front wheel	$t_{1,a} = 0$	$t_{1,l} = \frac{d}{v}$
Vehicles back wheel	$t_{2,a} = \frac{b_1 + b_2}{v}$	$t_{2,l} = \frac{b_1 + b_2 + d}{v}$
Trailers front wheel	$t_{3,a} = \frac{b_1 + l_1 + b_3}{v}$	$t_{3,l} = \frac{b_1 + l_1 + b_3 + d}{v}$
Trailers back wheel	$t_{4,a} = \frac{b_1 + l_1 + b_4}{v}$	$t_{4,l} = \frac{b_1 + l_1 + b_4 + d}{v}$

Figure 5. Times at approaching and leaving the bump

To make the input functions act with respect to the times when the wheel pass through the bump we have created following functions:

$$U_1(t) = 1 \rightarrow if \quad t_{1,l} > t \geq t_{1,a} \rightarrow else\ U_1(t) = 0$$
$$U_2(t) = 1 \rightarrow if \quad t_{2,l} > t \geq t_{2,a} \rightarrow else\ U_1(t) = 0$$
$$U_3(t) = 1 \rightarrow if \quad t_{3,l} > t \geq t_{3,a} \rightarrow else\ U_1(t) = 0$$
$$U_4(t) = 1 \rightarrow if \quad t_{4,l} > t \geq t_{4,a} \rightarrow else\ U_1(t) = 0$$

With the help of previous functions we can define the input functions in following way where *h* stands for the height of the bump and *d* stands for the length of the bump

$$z_1 = U_1(t) \cdot h \cdot \sin\left[\left(\frac{\pi \cdot v}{d}\right) \cdot t\right]$$
$$z_2 = U_2(t) \cdot h \cdot \sin\left[\left(\frac{\pi \cdot v}{d}\right) \cdot (t - t_{2,a})\right]$$
$$z_3 = U_3(t) \cdot h \cdot \sin\left[\left(\frac{\pi \cdot v}{d}\right) \cdot (t - t_{3,a})\right]$$
$$z_4 = U_4(t) \cdot h \cdot \sin\left[\left(\frac{\pi \cdot v}{d}\right) \cdot (t - t_{4,a})\right]$$

To build the forces resulting from damping that are proportional to the velocities, derivatives of input functions will be required:

$$\dot{z}_1 = U_1(t) \cdot h \cdot \left(\frac{\pi \cdot v}{d}\right) \cdot \cos\left[\left(\frac{\pi \cdot v}{d}\right) \cdot t\right]$$
$$\dot{z}_2 = U_2(t) \cdot h \cdot \left(\frac{\pi \cdot v}{d}\right) \cdot \cos\left[\left(\frac{\pi \cdot v}{d}\right) \cdot (t - t_{2,a})\right]$$
$$\dot{z}_3 = U_3(t) \cdot h \cdot \left(\frac{\pi \cdot v}{d}\right) \cdot \cos\left[\left(\frac{\pi \cdot v}{d}\right) \cdot (t - t_{3,a})\right]$$
$$\dot{z}_4 = U_4(t) \cdot h \cdot \left(\frac{\pi \cdot v}{d}\right) \cdot \cos\left[\left(\frac{\pi \cdot v}{d}\right) \cdot (t - t_{4,a})\right]$$

Figure 6. represents the input functions and their derivatives with respect to time:

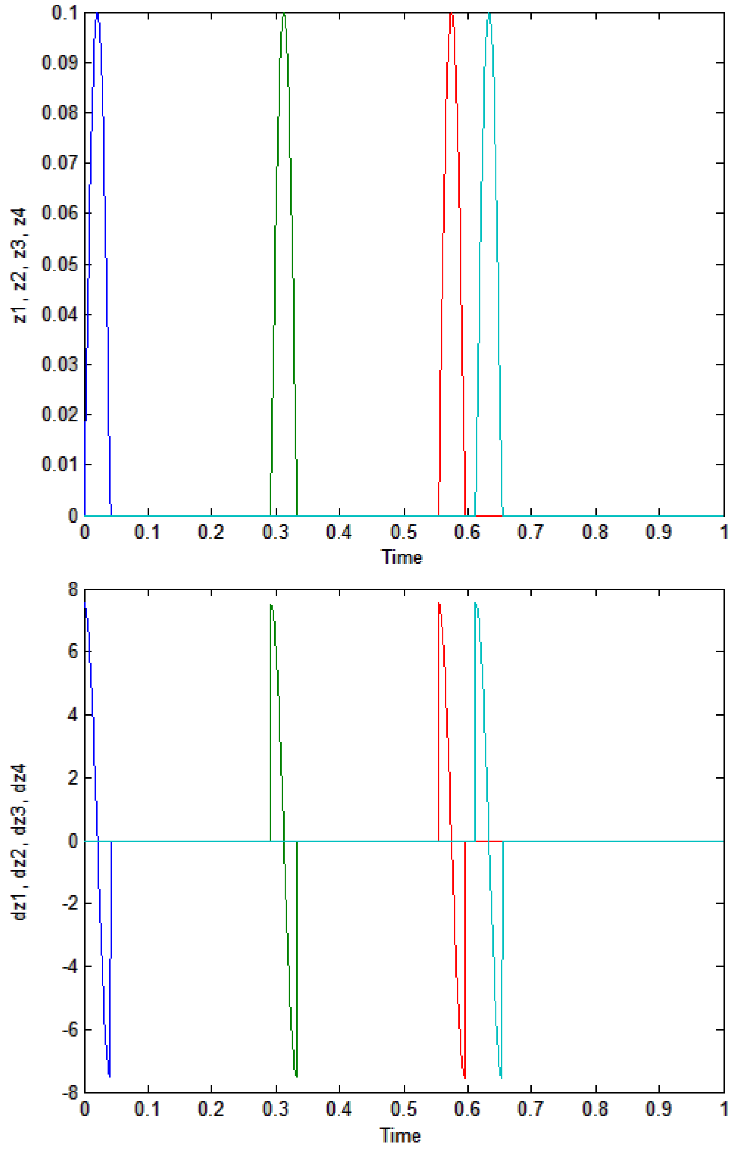

Figure 6. Input functions (up) and their derivatives (down) with respect to time

5.3. Assignment of the input functions to the degrees of freedom

Virtual work performed on virtual displacements may be written in following way:

$$\delta W = k_1 z_1 \delta a_1 + k_2 z_2 \delta a_2 + k_3 z_3 \delta a_3 + k_4 z_4 \delta a_4 + c_1 \dot{z}_1 \delta a_1 + c_2 \dot{z}_2 \delta a_2 + c_3 \dot{z}_3 \delta a_3 + c_4 \dot{z}_4 \delta a_4$$

$$\delta a_1 = \delta y_1 - \delta \varphi_1 b_1$$

$$\delta a_2 = \delta y_1 + \delta \varphi_1 b_2$$

$$\delta a_3 = \delta y_1 + \delta \varphi_1 l_1 + \delta \varphi_2 b_3$$

$$\delta a_4 = \delta y_1 + \delta \varphi_1 l_1 + \delta \varphi_2 b_4$$

Introducing the relations for virtual displacements into relation for virtual work and after algebraic operations we obtain:

$$\delta W = \delta y_1(k_1 z_1 + k_2 z_2 + k_3 z_3 + k_4 z_4 + c_1 \dot{z}_1 + c_2 \dot{z}_2 + c_3 \dot{z}_3 + c_4 \dot{z}_4)$$
$$+\delta\varphi_1(-k_1 b_1 z_1 + k_2 b_2 z_2 + k_3 l_1 z_3 + k_4 l_1 z_4 - c_1 b_1 \dot{z}_1 + c_2 b_2 \dot{z}_2 + c_3 l_1 \dot{z}_3 + c_4 l_1 \dot{z}_4) + \delta\varphi_2(k_3 b_3 z_3 +$$
$$k_4 b_4 z_4 + c_3 b_3 \dot{z}_3 + c_4 b_4 \dot{z}_4)$$

From previous equation is obvious that resulting force vector may be written as:

$$\boldsymbol{F} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} k_1 z_1 + k_2 z_2 + k_3 z_3 + k_4 z_4 + c_1 \dot{z}_1 + c_2 \dot{z}_2 + c_3 \dot{z}_3 + c_4 \dot{z}_4 \\ -k_1 b_1 z_1 + k_2 b_2 z_2 + k_3 l_1 z_3 + k_4 l_1 z_4 - c_1 b_1 \dot{z}_1 + c_2 b_2 \dot{z}_2 + c_3 l_1 \dot{z}_3 + c_4 l_1 \dot{z}_4 \\ k_3 b_3 z_3 + k_4 b_4 z_4 + c_3 b_3 \dot{z}_3 + c_4 b_4 \dot{z}_4 \end{bmatrix}$$

Finally our equations of motion in matrix form are:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{F}, \qquad \ddot{\boldsymbol{q}} = \begin{bmatrix} \ddot{y}_1 \\ \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \end{bmatrix}, \qquad \dot{\boldsymbol{q}} = \begin{bmatrix} \dot{y}_1 \\ \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix}, \qquad \boldsymbol{q} = \begin{bmatrix} y_1 \\ \varphi_1 \\ \varphi_2 \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

5.4. State space formulation

By expressing the acceleration vector from equation of motion in matrix form we obtain:

$$\ddot{\boldsymbol{q}} = -\boldsymbol{M}^{-1}\boldsymbol{C}\dot{\boldsymbol{q}} - \boldsymbol{M}^{-1}\boldsymbol{K}\boldsymbol{q} + \boldsymbol{M}^{-1}\boldsymbol{F},$$
$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}$$

State space formulation may be written in a form:

$$\begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I} \\ -\boldsymbol{M}^{-1}\boldsymbol{K} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1} \end{bmatrix} \boldsymbol{F} \rightarrow$$

$$\rightarrow \begin{bmatrix} \dot{y}_1 \\ \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \ddot{y}_1 \\ \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \\ \multicolumn{3}{c}{-\boldsymbol{M}^{-1}\boldsymbol{K}} & \multicolumn{3}{c}{-\boldsymbol{M}^{-1}\boldsymbol{C}} \\ \\ \end{bmatrix} \begin{bmatrix} y_1 \\ \varphi_1 \\ \varphi_2 \\ \dot{y}_1 \\ \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \\ & \boldsymbol{M}^{-1} & \\ \\ \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

$$\boldsymbol{q} = [\boldsymbol{I} \quad \boldsymbol{0}] \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix} + \boldsymbol{0F} \rightarrow$$

$$\rightarrow \begin{bmatrix} y_1 \\ \varphi_1 \\ \varphi_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ \varphi_1 \\ \varphi_2 \\ \dot{y}_1 \\ \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

5.5. Simulation in Matlab

State space matrixes were build and simulation for one second with 1000 steps was performed. Used script is included in appendix. Figure 7. represents the results of the rotations and displacement of the degrees of freedom:

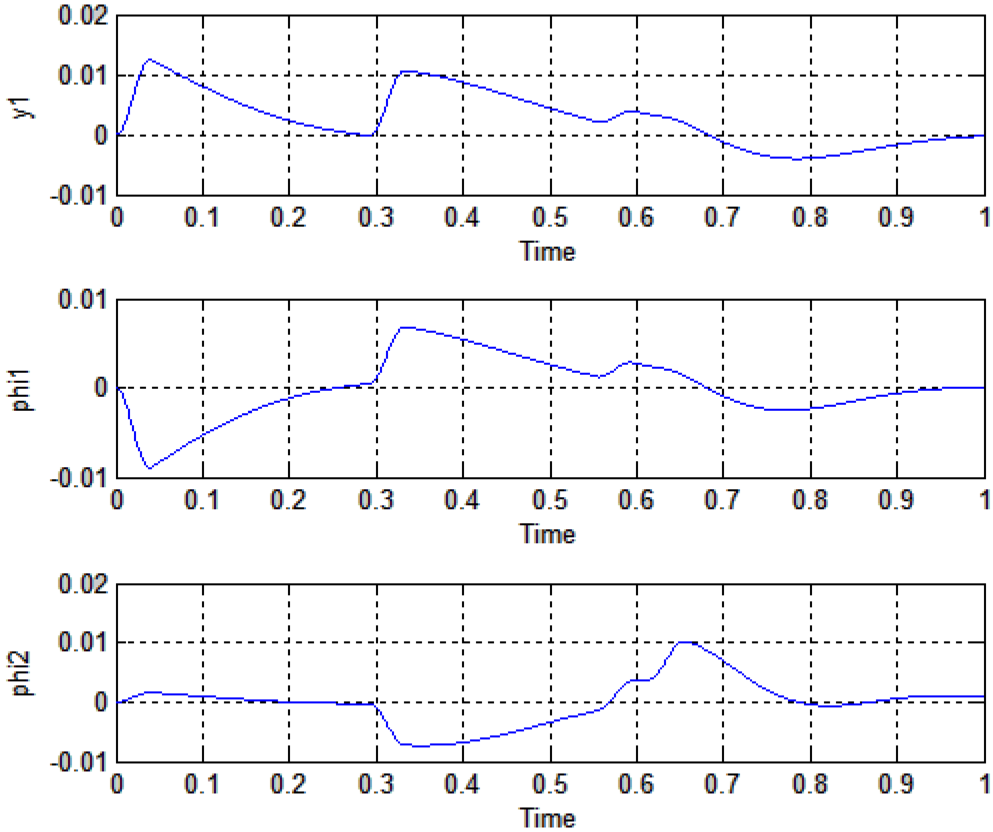

Figure 7. Displacements and rotations of the degrees of freedom

Figure 8. shows the spring deflections with respect to time and the displacement of the trailers centre of the gravity.

Spring deflections and displacement of the trailers centre of the gravity were computed by following expressions:

$$y_2 = y_1(t) + \varphi_1(t) \cdot l_1 + \varphi_2(t) \cdot l_2$$

$$a_1 = y_1(t) - \varphi_1(t) \cdot b_1$$

$$a_2 = y_1(t) + \varphi_1(t) \cdot b_2$$

$$a_3 = y_1(t) + \varphi_1(t) \cdot l_1 + \varphi_2(t) \cdot b_3$$

$$a_4 = y_1(t) + \varphi_1(t) \cdot l_1 + \varphi_2(t) \cdot b_4$$

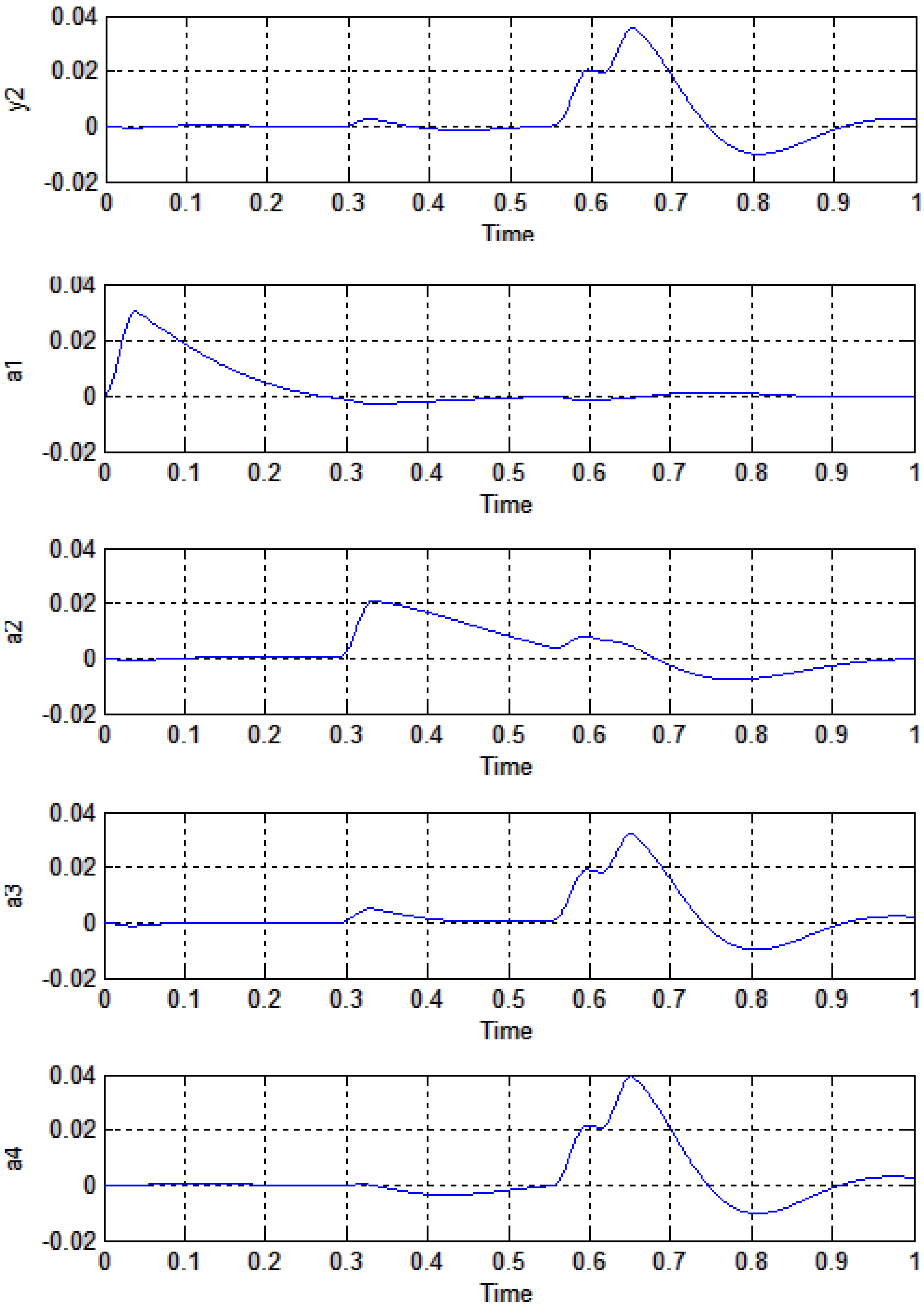

Figure 7. Displacement of the trailers centre of the gravity and deflections of the springs

6. Road input definition and model response simulation in MSC.adams

Our modelling approach in MSC.adams was straight forward. To model a bump we took the points generated in Matlab (points that creates figure 4.) and put them into the MSC.adams as a construction pints.

Addition construction point were build to create flat road behind and in front of the bump. All the construction points were connected by poly-line what created new part.

Springs were attached to the ends of the mass less parts. Other ends of the mass less parts were connected to the poly-line by point to curve constrain. Finally the mass less parts were attached to the ground by vertical translational joint. Their purpose is to transmit the kinematic excitations into the model.

The poly-line part was connected to ground by horizontal translation joint and to this joint was prescribed the translational joint motion with velocity equal to the velocity used in Matlab simulation.

The damping values were prescribed for each spring according to the values used in Matlab simulation.

Duration of the simulation was set to one second and time step was set to 1000 steps.

Simulation was started from static equilibrium.

Two measures were created. First represents the vertical displacement of the vehicles centre of the gravity and second represents the vertical displacement of the trailers centre of the gravity.

Figure 8. represents the model used for simulation with description of each components and figure 9. represents the results of the measures.

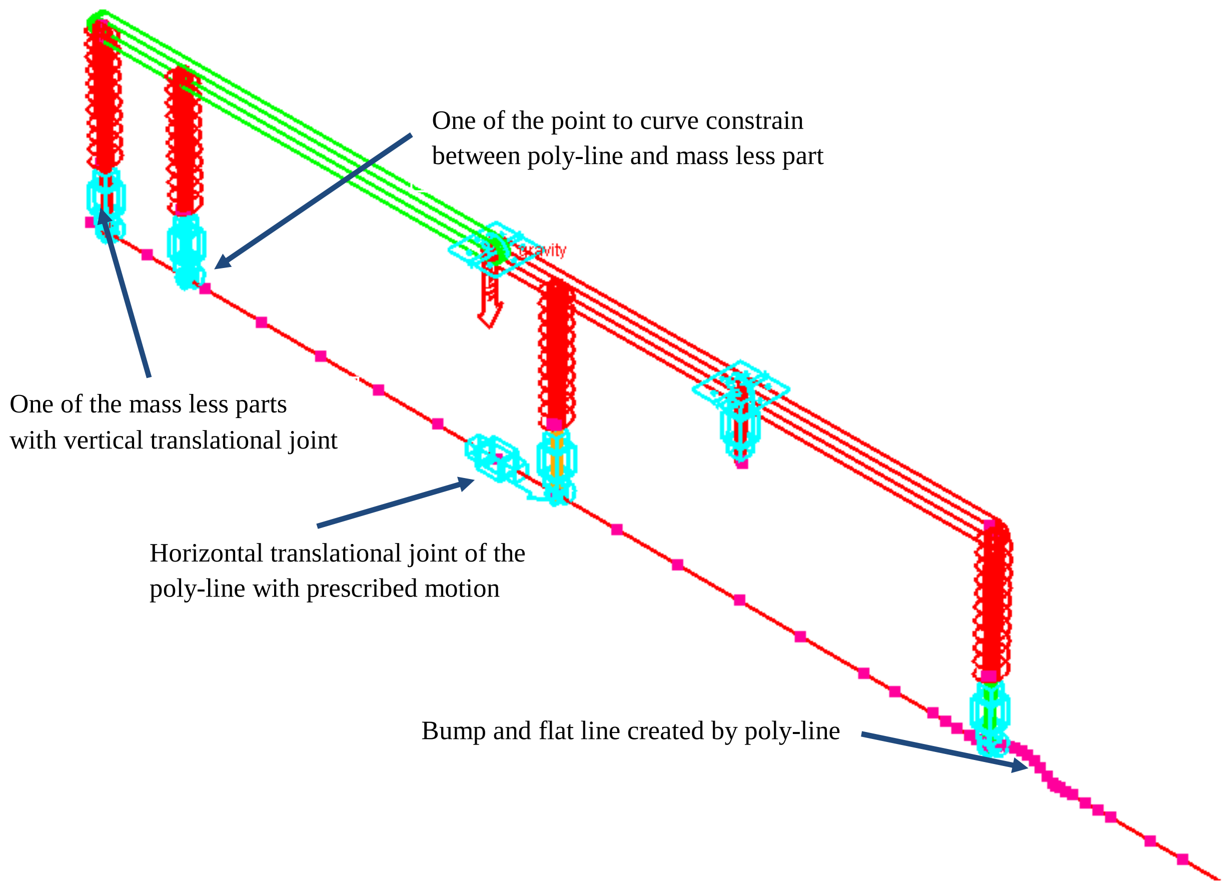

Figure 8. Model used for simulation in MSC.adams

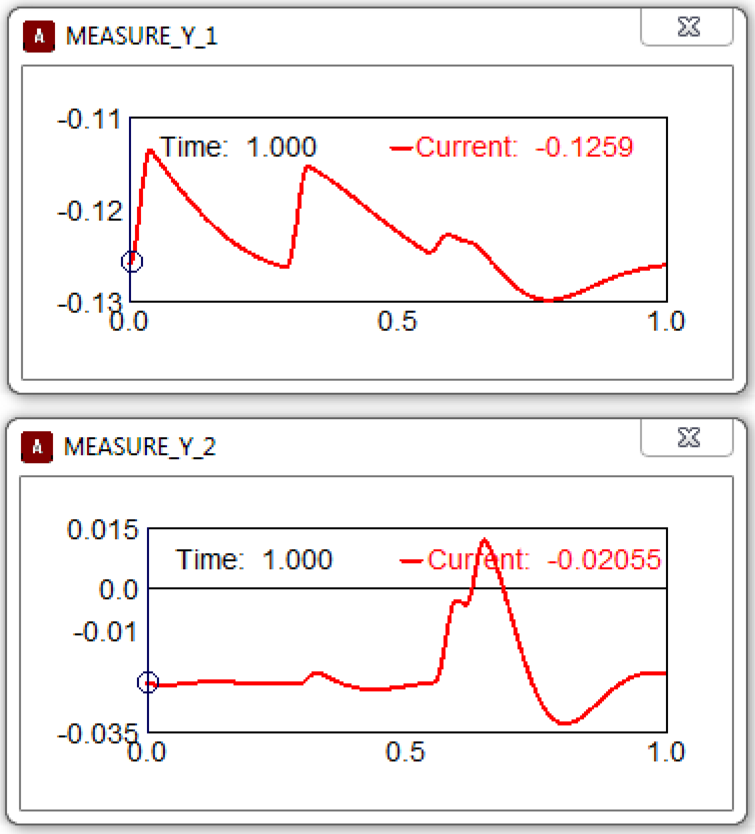

Figure 9. Results from simulation performed in MSC.adams

7. Comparison of the model response results.

Results from MSC.adams were generated and imported into Matlab. Results from MSC.adams have a certain offset due to the start from static equilibrium what caused springs to deflect. This offset was deleted and results were plotted in one graph. This graph represents figure 10.:

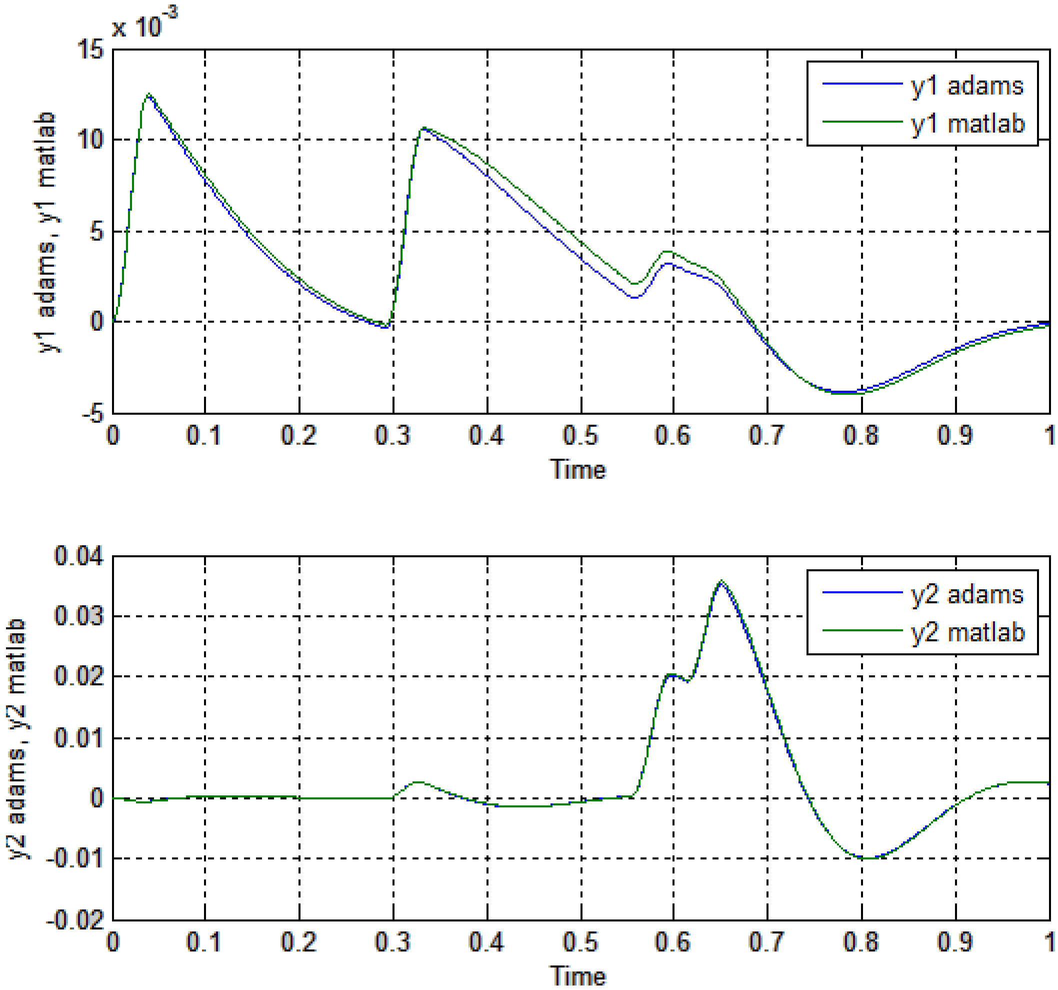

Figure 10. Comparison of the simulation results.

From the figure 10 is visible that results are matched very accurately. That indicates that our modelling strategy including the definition of the input function, state space formulation and other factors was correct. Figure 10. shows only the centre of the gravity displacements comparison. Displacement of the trailers centre of the gravity is computed by relation between all three degrees of freedom. Fact that the displacement of the trailers centre of the gravity is correctly computed according to the comparison with MSC.adams results, implies that results for rotational degrees of freedom are correct as well.

Summary

Equations of motion were derived directly from relations for kinetic and potential energy. Their correctness was validated by comparison of natural frequencies computed in Matlab and in MSC.adams. Results matched with very small difference.

Other task was to simulate the transition of the vehicle with trailer trough the bump. Bump was modelled as a half sine. Input functions were build and simulation performed.

Alternative model was created in MSC.adams with attempt to imitate the conditions used in Matlab model. After comparison of the simulations performed in different environments, results matched again with very small difference.

By taking the results from MSC.adams as a reference results, is our modelling strategy confirmed as a correct what gives us an engineering satisfaction.

Appendix

Matlab script:

```
clc
clf
clear all

%% Parameters
L1=2;           %length dimension (see report for figure)
L2=3;
mi1=2;
mi2=1.5;
mi3=2.65;
mi4=3.35;

m1=1200;        %mass of the vehicle
m2=700;         %mass of the trailer
I1=3000;        %moment of inertia of the vehicle
I2=1200;        %moment of inertia of the trailer

k1=42000;       %spring stiffness
k2=48000;
k3=160000;
k4=160000;

c1=6100;        %viscous damping coefficient
c2=6400;
c3=6000;
c4=6000;

%% Mass matrix
m11=m1+m2;
m12=m2*L1;
m13=m2*L2;
m22=m2*L1*L1+I1;
m23=m2*L1*L2;
m33=m2*L2*L2+I2;
M=[m11 m12 m13;m12 m22 m23;m13 m23 m33];

%% Stiffness matrix
k11=k1+k2+k3+k4;
k12=-k1*mi1+k2*mi2+k3*L1+k4*L1;
k13=k3*mi3+k4*mi4;
k22=k1*mi1*mi1+k2*mi2*mi2+k3*L1*L1+k4*L1*L1;
k23=k3*L1*mi3+k4*L1*mi4;
k33=k3*mi3*mi3+k4*mi4*mi4;
K=[k11 k12 k13;k12 k22 k23;k13 k23 k33];

%% Natural frequencies%
[PHI, LAMBDA]=eig(K,M);
f=sqrt(LAMBDA)/(2*pi);
f1=f(1,1);
f2=f(2,2);
f3=f(3,3);
```

```
%% ADAMS results%
f1A=1.286240;
f2A=1.517599;
f3A=3.197859;

%% Absolute difference%
miss1=abs(f1-f1A);
miss2=abs(f2-f2A);
miss3=abs(f3-f3A);

%% Mode shapes visualization
phi_1=PHI(:,1);
phi_2=PHI(:,2);
phi_3=PHI(:,3);

figure(1)
plot(phi_1,'- .' ,'LineWidth',2,'MarkerSize',20)
grid on
hold on
plot(phi_2,'- .g','LineWidth',2,'MarkerSize',20)
grid on
hold on
title('Mode shapes')
plot(phi_3,'- .r','LineWidth',2,'MarkerSize',20)
legend('Mode 1','Mode 2','Mode 3')
grid on
hold on

%% Damping matrix
c11=c1+c2+c3+c4;
c12=-c1*mi1+c2*mi2+c3*L1+c4*L1;
c13=c3*mi3+c4*mi4;
c22=c1*mi1*mi1+c2*mi2*mi2+c3*L1*L1+c4*L1*L1;
c23=c3*L1*mi3+c4*L1*mi4;
c33=c3*mi3*mi3+c4*mi4*mi4;
CC=[c11 c12 c13;c12 c22 c23;c13 c23 c33];

%% State space matrixes
OsA=zeros(size(M));
Is=eye(size(M));
A=[OsA Is;-M^(-1)*K -M^(-1)*CC];
B=[OsA;M^(-1)];
C=[Is OsA];
D=zeros(size(C,1),size(B,2));

%% Input functions
t=0:1/999:1;                                  %time interval

h0=0.1;                                       %height of the bump
d=0.5;                                        %length of the bump
v=12;                                         %velocity
omg=(pi*v)/d;

t2_start=(mi1+mi2)/v;                  %time at approaching the bump (2.wheel)
t3_start=(mi1+L1+mi3)/v;               %time at approaching the bump (3.wheel)
t4_start=(mi1+L1+mi4)/v;               %time at approaching the bump (4.wheel)
```

```
z1=U1(t).*h0.*sin(t*omg);                     %input functions
z2=U2(t).*h0.*sin((t-t2_start)*omg);
z3=U3(t).*h0.*sin((t-t3_start)*omg);
z4=U4(t).*h0.*sin((t-t4_start)*omg);

figure(2)
plot(t,z1,t,z2,t,z3,t,z4)
xlabel('Time')
ylabel('z1, z2, z3, z4')

dz1=U1(t).*omg.*h0.*cos(t*omg);               %derivatives of input functions
dz2=U2(t).*omg.*h0.*cos((t-t2_start)*omg);
dz3=U3(t).*omg.*h0.*cos((t-t3_start)*omg);
dz4=U4(t).*omg.*h0.*cos((t-t4_start)*omg);

figure(3)
plot(t,dz1,t,dz2,t,dz3,t,dz4)
xlabel('Time')
ylabel('dz1, dz2, dz3, dz4')

%% Forces assigned to the DOFs
f1=k1*z1+k2*z2+k3*z3+k4*z4+c1*dz1+c2*dz2+c3*dz3+c4*dz4;
f2=-mi1*k1*z1+mi2*k2*z2+L1*k3*z3+L1*k4*z4-
mi1*c1*dz1+mi2*c2*dz2+L1*c3*dz3+L1*c4*dz4;
f3=mi3*k3*z3+mi4*k4*z4+mi3*c3*dz3+mi4*c4*dz4;

F=[f1;f2;f3];

%% Simulation and results in time domain
sys=ss(A,B,C,D);
y=lsim(sys,F,t);

figure(4)
subplot(4,1,1)
plot(t,y(:,1))
grid
xlabel('Time')
ylabel('y1')
subplot(4,1,2)
plot(t,y(:,2))
grid
xlabel('Time')
ylabel('phi1')
subplot(4,1,3)
plot(t,y(:,3))
grid
xlabel('Time')
ylabel('phi2')

y2=y(:,1)+L1.*y(:,2)+L2.*y(:,3);   %Trailers centre of gravity displacement

subplot(4,1,4)
plot(t,y2)
grid
xlabel('Time')
ylabel('y2')
```

```
a1=y(:,1)-mi1.*y(:,2);              %Deflection of the springs
a2=y(:,1)+mi2.*y(:,2);
a3=y(:,1)+L1.*y(:,2)+mi3.*y(:,3);
a4=y(:,1)+L1.*y(:,2)+mi4.*y(:,3);


figure(5)
subplot(4,1,1)
plot(t,a1)
grid
xlabel('Time')
ylabel('a1')
subplot(4,1,2)
plot(t,a2)
grid
xlabel('Time')
ylabel('a2')
subplot(4,1,3)
plot(t,a3)
grid
xlabel('Time')
ylabel('a3')
subplot(4,1,4)
plot(t,a4)
grid
xlabel('Time')
ylabel('a4')

%% Comparison with Adams results

data_1=load('ADAMS_Y_1.tab');     % Adams data import (see included data)
t_ad=data_1(:,1);
y1_ad=data_1(:,2);
data_2=load('ADAMS_Y_2.tab');
y2_ad=data_2(:,2);

%% Correction of results from Adams due to static compression of springs

delta_y1=y(1,1)-y1_ad(1,1);
y1_ad_red=y1_ad+delta_y1;

delta_y2=y2(1,1)-y2_ad(1,1);
y2_ad_red=y2_ad+delta_y2;


figure(6)
subplot(2,1,1)
plot(t_ad,y1_ad_red,t,y(:,1));
grid
xlabel('Time')
ylabel('y1 adams, y1 matlab')
legend('y1 adams','y1 matlab')
subplot(2,1,2)
plot(t_ad,y2_ad_red,t,y2);
grid
xlabel('Time')
ylabel('y2 adams, y2 matlab')
legend('y2 adams','y2 matlab')
```

.m functions:

```
function u1 = U1(t)

d=0.5;
v=12;
t1_start=0;
t1_end=d/v;

u1=(t >= t1_start & t < t1_end);
end


function u2 = U2(t)
d=0.5;
mi1=2;
mi2=1.5;
v=12;

t2_start=(mi1+mi2)/v;
t2_end=(d+mi1+mi2)/v;

u2=(t >= t2_start & t < t2_end);
end


function u3 = U3(t)

d=0.5;
L1=2;
mi1=2;
mi3=2.65;
v=12;

t3_start=(mi1+L1+mi3)/v;
t3_end=(d+mi1+L1+mi3)/v;

u3=(t >= t3_start & t < t3_end);
end


function u4 = U4(t)

d=0.5;
L1=2;
mi1=2;
mi4=3.35;
v=12;

t4_start=(mi1+L1+mi4)/v;
t4_end=(d+mi1+L1+mi4)/v;

u4=(t >= t4_start & t < t4_end);
end
```