# A SYMBOLIC APPROACH TO THE MULTI-BODY MODELLING OF ROAD VEHICLES

ROBERTO LOT
*Faculty of Engineering and the Environment, University of Southampton*
*University Road, Southampton SO17 1BJ, UK*

MATTEO MASSARO
*Department of Industrial Engineering, University of Padova*
*Via Venezia 1, 31015 Padova, IT*

Received January 3, 2017
Accepted May 29, 2017

This paper introduces *MBSymba*, an object-oriented language for the modelling of multi-body systems and the automatic generation of equations of motion in symbolic form. *MBSymba* has built upon the general-purpose computer algebra software Maple and it is freely available for teaching and research purposes. With *MBSymba*, objects such as points, vectors, rigid bodies, forces and torques, and the relationships among them may be defined and manipulated both at high and low levels. Absolute, relative or mixed coordinates may be used, as well as combination of infinitesimal and non-infinitesimal variables. Once the system has been modelled, Lagrange's and/or Newton's equations can be derived in a quasi-automatic way, either in an inertial or non-inertial reference frame. Equations can be automatically converted into Matlab, C/C++ or Fortan code to produce stand alone, numerically optimized simulation code. *MBSymba* is particularly suited for the modelling of ground, water or air vehicles; therefore, the mathematical model of a passenger car with trailer is illustrated as a case study. Time domain simulations, steady state analysis and stability results are also presented.

*Keywords*: Symbolic Multibody; Vehicle Dynamics and Control

## 1. Introduction

Nowadays, the most popular software tools for the modelling of multibody systems are geared towards product design and offer detailed modelling and simulation of mechanical systems, e.g. ADAMS Li and Kota (2001); MSC (2016), Virtual.Lab Motion Siemens (2016) (formerly released as DADS), SIMPACK Simulia (2016), and Simwise Wang (2001); DST (2016) (formerly released as VisualNastran/WorkingModel). These are based on a numerical approach and typically use a large number of variables, much greater than the number of degrees of freedom.

However, for some classes of applications (such as preliminary and conceptual design Massaro *et al.* (2011), control design Biral *et al.* (2010), minimum-time problems Tavernini *et al.* (2013); Lot and Bianco (2015), real-time simulation Cossalter *et al.* (2011b) and embedded systems), it is still more attractive to develop "essential" and "smart" mathematical models that capture the most important features of the system and that may be inspected and manipulated at symbolic level. In addition, the symbolic models can be clearly exported towards different environments, thus portability is another significant advantage related to the symbolic approach.

For this reasons, a number of packages have been introduced along the years to efficiently generate the equations of motion for multibody systems, either as add-ons to general-purpose computer algebra software or as stand alone programs, e.g. MotionGenesis Levinson (1977); Schiehlen (1990); Motion Genesis (2016) (released as Autolev until 2010), Neweul-$M^2$ (originally released as Neweul) Kreuzer (1979); Schiehlen (1990); Kurz *et al.* (2010); Stuttgart University (2016), VehicleSim Lisp or VS Lisp (previously released as Autosim Sayers (1990, 1991, 1993) and now distributed in specialised versions such as CarSim, TruckSim and BikeSim Mechanical Simulation (2016)), SD/FAST Rosenthal (1988); PTC (2016), Robotran Schiehlen (1990); Samin and Fisette (2003); Docquier *et al.* (2013), and MapleSim, which started as DynaFlexPro Shi and McPhee (2002); Maplesoft (2016).

This paper describes *MBSymba*, an object-oriented language for the modelling of multibody systems and the automatic symbolic generation of the related equations of motion, which employs the symbolic mathematics kernel of Maple Maplesoft (2016). The main feature of the software, when compared with those previously mentioned, is the flexibility given to the user in the modelling phase. While a basic version of *MBSymba* was released a decade ago Lot and Da Lio (2004), the updated version presented here includes major improvements, such as a moving frame approach and "smart" linear modelling features. These are particularly useful when developing multibody models of ground vehicles, as well as watercraft and aircraft.

The modelling procedure in *MBSymba* requires the user to form a description of the multibody system by defining objects such as points and vectors, rigid bodies, forces and torques and the kinematic relationships among them. At this stage, the user has total freedom in the choice of generalized coordinates, which is essential for building small and effective models. It is possible to use either absolute or relative coordinates, or a combination of both, to give either a dependent or independent formulation. At this level, vectors and points may be defined and manipulated according to classical geometric rules, while symbolic expressions for speed and acceleration are automatically derived and projected onto any desired reference frame. The definition and manipulation of bodies, forces and torques follow the rules of classical mechanics. Symmetries and other special properties of the system especially benefit from the symbolic approach, as any model parameter vanishes when it is declared zero. Land vehicles, as well as watercraft and aircraft, whose dynamics are typically not affected by their position and orientation with respect to a fixed reference frame, can be efficiently modelled using a moving, non-inertial base reference frame. This option, as well as the automatic management of cyclic coordinates, is available in *MBSymba*.

Another important feature is the possibility of easily and properly modelling partially linear systems, i.e. systems where some coordinates and parameters are small, while others are not. For such systems the user has to declare the small variables and *MBSymba* will automatically perform the appropriate simplifications at each modelling step.

Once the system is modelled, the derivation of the equations of motion can be performed in a quasi-automatic way: the user retains the freedom, as well as the responsibility, to select a consistent set of equations of motion, but the actual calculation of the equations of motion is performed by *MBSymba* and Maple according to the system description. Both Lagrange's or Newton's approach are available. Lagrange's approach is very popular since it gives a minimum set of equations by differentiation of scalar quantities such as kinetic energy, potential energy and virtual work and can easily deal with constraints. However Newton's formulation may be a better choice when modelling linear systems, and sometimes gives more compact equations (e.g. Euler's equations for a rotating body).

Additionally, the user can calculate, inspect and manipulate intermediate objects' properties, such as potential and kinetic energy, linear and angular momentum for bodies, power and generalized forces for forces and torques, etc. Finally, model equations as well as intermediate results can be converted into Matlab, C or Fortran code to produce stand alone, numerically optimized simulation programs.

A version of MBSymba is available on the web `www.multibody.net`, with a guide and several examples.

## 2. Definition of the Multibody System

To illustrates *MBSymba* capabilities, a symbolic multibody model of a passenger car with trailing trailer is developed, cornering performance and stability are also assessed. Model characteristics and degrees of freedom are sketched in figure 1. It is assumed that tyres are rigid, therefore wheels and other car unsprung masses may be modelled as a sole *unsprung* rigid body which has three degrees of freedom: the longitudinal and lateral speed $u, v$ and the yaw rate $r$. The *chassis* is suspended on the unsprung mass by a system consisting of four vertical spring-damper elements: the additional three degrees of freedom are the bounce $z$, roll $\phi$ and pitch $\mu$. Therefore the car has six degrees of freedom in total. The *trailer* has no suspensions and it is modelled as a single rigid body, whose motion is described by means of the coordinates $x_t, y_t, z_t$ of its gravity centre $\mathbf{T}$, the yaw angle $\alpha$ relative respect to the car and pitch rotation $\beta$. The roll motion is not included since it is impeded by the tyres, therefore the trailer has four degrees of freedom in total. When the trailer is connected to the car by the hitch trailer, spherical joint $\mathbf{H}$, it looses three degrees of freedom and in conclusion the car-trailer system has seven degrees of freedom, while its kinematics is described with a set of ten dependent coordinates, namely $u, v, r, z, \phi, \mu, x_t, y_t, z_t, \alpha, \beta$.

In the next sections, the multibody system is built up step by step and at the same time the principal characteristics of *MBSymba* are illustrated, including the definition and manipulations of frames, points and vectors, rigid bodies, forces and

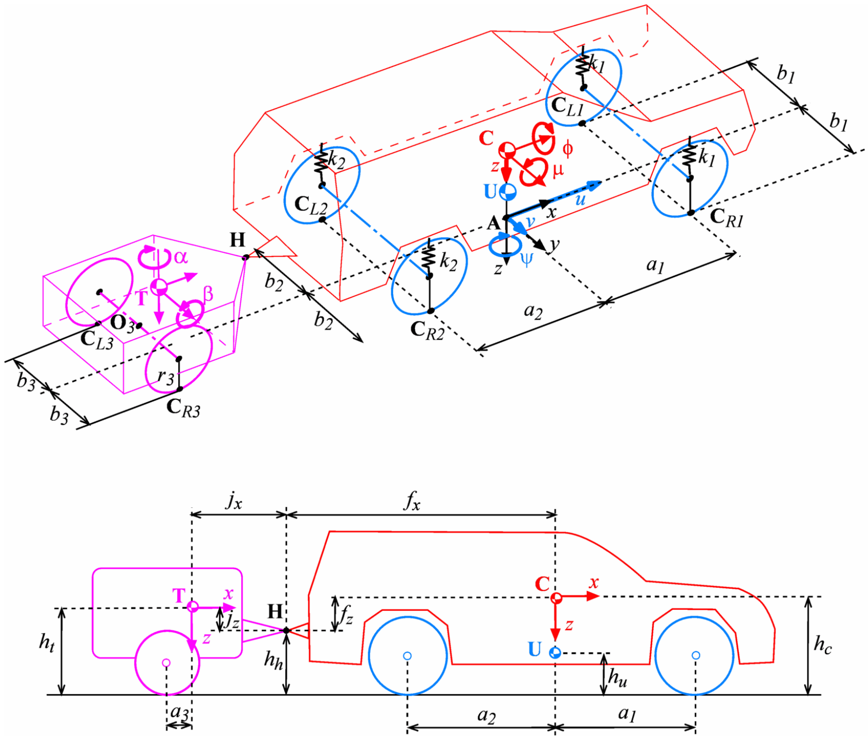

Fig. 1. Car trailer model.

torques, and constraints. Furthermore, the mix of infinitesimal and non-infinitesimal variables will be discussed and the advantages of the moving frame highlighted.

### 2.1. *Frames, points and vectors*

The fundamental building blocks of *MBSymba* are cartesian reference *frames*; other objects such as points, bodies and forces need be defined with respect to a reference frame. The only pre-defined object in *MBSymba* is the inertial frame, which is called *ground*. In *MBSymba*, a reference frame $j$ can be defined by providing the coordinates of its origin and the direction cosines of its axes with respect to another reference frame $i$, and this information can be collected into a square $4 \times 4$ matrix $\mathbf{T}_i^j$ Suh and Radcliffe (1978), where the $3 \times 3$ upper left block is a rotation sub-matrix, the $3 \times 1$ upper right block contains the coordinates of the origin, while $1 \times 1$ bottom right is always 1. Such matrix can also be associated to the rigid body motion which moves the frame $i$ to $j$. Hence this $4 \times 4$ matrix formulation makes it possible to easily concatenate sequences of transformations, i.e. the rigid motion from the frame $i$ to a frame $k$ can be written as the rigid motion from $i$ to $j$ and from $j$ to $k$ as follows:

$$\mathbf{T}_i^k = \mathbf{T}_i^j \, \mathbf{T}_j^k \tag{1}$$

In *MBSymba* transformation matrices are easily defined by combining translations and rotations with respect to the coordinate axes. In our model, the car gross motion may be tracked by defining a reference frame having its origin below the point $\mathbf{U}$ at ground level and whose $x$ axis points forward:

$$\mathbf{T}_0 = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & x \\ \sin\psi & \cos\psi & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where $x$ and $y$ are the (time dependent) translation with respect to the *ground* frame and $\psi$ is the (time dependent) yaw rotation.

Points and vectors are created by defining their cartesian components in a certain reference frame, for instance the centre of mass of the car unsprung masses $\mathbf{U}$ may be defined as follows:

$$\mathbf{U} = \begin{pmatrix} 0 \\ 0 \\ -h_u \\ 1 \end{pmatrix}_0 \tag{3}$$

Where the suffix $_0$ highlight that point coordinates are expressed in the reference frame $\mathbf{T}_0$. Point coordinates and frame descriptions are embedded into a single object in order to avoid any possible misinterpretation. Note also that the point components are stored in a 4 row array, where the first three rows consist of the $x$, $y$ and $z$ components with respect to the specified frame, while the fourth row is always 1 in the case of points and 0 in the case of vectors.

### 2.2. *Inertial and non-inertial bases*

For vehicles (and in many other cases) it is convenient to describe the multibody system with respect to a non-inertial moving reference frame instead of the *ground*. Indeed, the dynamics of a vehicle running on a flat road does not depend on its position nor on its orientation (they are invariant by translation and rotation on the road plane), hence specifying the transformation matrix (2) would be redundant and inconvenient. To take advantage of such situations, *MBSymba* allows to replace the *ground* frame with a moving, non-inertial reference frame, specified by the user. This reference frame is defined in terms of the absolute velocity of its origin $\mathbf{V}_0 = (u, v, w)^T$ and its absolute angular velocity $\mathbf{\Omega} = (\Omega_x, \Omega_y, \Omega_y)^T$, both expressed with respect to the moving reference frame itself. For the proposed case study, it is convenient to derive the model with respect to the moving frame $\mathbf{T}_0$ previously defined. When defining a multibody model with respect to a non-inertial reference frame, it is not strictly necessary to define the position of the moving frame with respect to the ground frame. Definition (2) could be omitted, because frame velocity and acceleration matter, while position and orientation do not. In any case, it is possible to track the vehicle by specifying the position of the moving frame (which has velocities $\mathbf{V}_0$ and $\mathbf{\Omega}$) with respect to the ground. Tracking equations are automatically generated:

$$\begin{aligned} u &= \dot{x}\cos\psi + \dot{y}\sin\psi \\ v &= -\dot{x}\sin\psi + \dot{y}\cos\psi \\ r &= \dot{\psi} \end{aligned} \tag{4}$$

and, by using Maple's capabilities, they may be easily rearranged into a set of ordinary differential equations (ODE):

$$\begin{aligned} \dot{x} &= u\cos\psi - v\sin\psi \\ \dot{y} &= u\sin\psi + v\cos\psi \\ \dot{\psi} &= r \end{aligned} \tag{5}$$

which give the vehicle trajectory $x, y, \psi$ by integration. Once the moving frame has been declared, from now on *MBSymba* uses the following rule to calculate the time derivative of any vectorial object $\mathbf{A}$:

$$\frac{d}{dt}\mathbf{A} = \mathbf{\Omega} \times \mathbf{A} + \dot{\mathbf{A}} \tag{6}$$

or, in matrix form:

$$\mathbf{V}_A = \mathbf{W}\mathbf{A} + \dot{\mathbf{A}} \tag{7}$$

where the velocity matrix $\mathbf{W}$ is defined as:

$$\mathbf{W} = \begin{bmatrix} 0 & -\Omega_z & \Omega_y & u \\ \Omega_z & 0 & -\Omega_x & v \\ -\Omega_y & \Omega_x & 0 & w \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

As an example, the velocity and acceleration of the unsprung centre of mass $\mathbf{U}$ are calculated as follows:

$$\mathbf{V}_U = \begin{pmatrix} u \\ v \\ 0 \\ 0 \end{pmatrix}, \qquad \mathbf{A}_U = \begin{pmatrix} -vr + \dot{u} \\ ur + \dot{v} \\ 0 \\ 0 \end{pmatrix}$$

It is worth pointing out that equation (7) is valid for any vectorial object (e.g. it is used to calculate Newton's equations as time derivative of linear and angular momentum) and may also be applied recursively. It is also obvious that the moving frame has to be declared at the beginning of the modelling phase and cannot be changed afterwards.

### 2.3. *Smart modelling of linear systems*

Even if the equations of motion of multibody systems are in general non-linear, often systems works on the proximity of a fixed configuration, or at least the variation of some variables remains small. In those cases, it is convenient to linearize the equations of motion with respect to such small variables. If a dynamic equilibrium configuration exists, the equations of motion may be even linearized with respect to all variables, therefore taking advantage of techniques such as fast computation of linear response, eigenvalue and frequency analysis, linear control design, and other tools that are not available for non-linear models. The linearization of the equations of motion obviously requires the prior knowledge of the non-linear equations. However, when dealing with complex non-linear expressions that typically arise in kinematic chains, the symbolic derivation of the non-linear equations of motions may be quite challenging and sometimes even infeasible.

In this context, *MBSymba* offers the possibility to completely avoid the necessity of considering non-linear expressions while modelling a linear system. *MBSymba* simply requires the specification of the list of variables that should be considered small. After that, all the expressions that contain such variables are automatically and consistently linearized, and any unnecessary non-linear expression is immediately discarded. For example, for both the steady cornering and stability analysis of the car-trailer system, it is reasonable to assume that the oscillations $z, \phi, \mu$ of the car chassis are small The reference frame $\mathbf{T}_C$ attached to the car chassis begins:

$$\mathbf{T}_C = \begin{bmatrix} 1 & 0 & \mu & 0 \\ 0 & 1 & -\phi & 0 \\ -\mu & \phi & 1 & z - h_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where $h_c$ is the gravity centre height in static conditions. Since trigonometric expressions have been automatically linearized by *MBSymba*, the transformation matrix $\mathbf{T}_C$ has the same structure of the velocity matrix $\mathbf{W}$ in equation (7). With the linear modelling option, rotations transformation are treated as infinitesimal quantities, in the sense that they are "orthogonal transformation of coordinate axes in which the component of a vector are almost the same in both sets of axes" (Goldstein *et al.* (2000), section 4.8) and this concept may be easily extended to small translations too.

While using the linear modelling option, particular caution must be taken in the calculation of derivatives, including the calculation of velocities and accelerations. When either a displacement or rotation variable $y$ depends on another position variable $x$, the dependent position, velocity and acceleration may be calculated as follows:

$$y = f(x) \tag{10a}$$
$$\dot{y} = f'(x)\dot{x} \tag{10b}$$
$$\ddot{y} = f'(x)\ddot{x} + f''(x)\dot{x}^2 \tag{10c}$$

Now, by assuming that the position $x$ is small, but its velocity $\dot{x}$ is not, the former equations may be linearized as follows:

$$y \simeq f(0) + f'(0)x \tag{11a}$$
$$\dot{y} \simeq f'(0)\dot{x} + f''(0)x\,\dot{x} \tag{11b}$$
$$\ddot{y} \simeq f'(0)\ddot{x} + f''(0)\dot{x}^2 + f''(0)x\,\ddot{x} + f'''(0)x\,\dot{x}^2 \tag{11c}$$

*MBSymba* immediately linearize any position expression in the form (11a). As a consequence, any additional information about $f(x)$ is lost and in particular the $f''(0)$ and $f'''(0)$ terms in equations (11) are lost. However, if $\dot{x}, \ddot{x}$ are small too, such additional terms become negligible and the linear modelling technique is correct.

Returning to the car-trailer example, not only variables $z, \phi, \mu$ may be considered small, but also their velocity $\dot{z}, \dot{\phi}, \dot{\mu}$ and accelerations $\ddot{z}, \ddot{\phi}, \ddot{\mu}$. The yaw acceleration $\dot{r}$ may be also considered small, even if the yaw rate $r$ is not. When tyre forces are safely below their saturation limit, the lateral speed $v$ and its time derivative $\dot{v}$ are small too. For the trailer, the relative yaw angle $\alpha$ cannot be considered small, while the pitch rotation $\beta$, as well as velocities $\dot{\alpha}, \dot{\beta}$ and accelerations $\ddot{\alpha}, \ddot{\beta}$ are small. This situation is reflected in *MBSymba*, where small displacements and rotations $(\phi, \mu, z, \beta)$, small velocities $(v, \dot{\alpha})$, and small acceleration $\dot{r}$ are declared separately. The following list of small variables $\sigma$ is generated:

$$\sigma = \Big(\beta, \phi, \mu, z, \dot{\alpha}, \dot{\beta}, \dot{\phi}, \dot{\mu}, \dot{z}, v, \dot{v}, \ddot{\alpha}, \ddot{\beta}, \ddot{\phi}, \ddot{\mu}, \ddot{z}, \dot{r}\Big) \tag{12}$$

In the next sections the advantages of such approach will be extensively discussed.

There are other situations when the linear modelling approach is not practicable. For example, in a high frequency vibration, even if the displacement $x = a \sin \omega t$ is small, the velocity $\dot{x} = (a\omega)\cos \omega t$ is not. In this case, as already discussed, the linear modelling option cannot be used. However, *MBSymba* can be still used to derive the non-linear equations of motion and linearize them at the end.

A more important limitation of the linear modelling approach is its incompatibility with the Lagrange's equations of motion. Indeed, the expressions of potential and kinetic energy of linear systems are quadratic, but, as already discussed, quadratic terms may be lost after premature linearization. Quadratic expressions are also necessary for the correct calculation of generalized forces, that after derivation with respect to the dependent variables are reduced to first order. In any case, it is worth pointing out that the compatibility between linear modelling and Lagrange's approach has to be evaluated variable by variable. In other words, a system may include both small and non small variables: Lagrange's equations cannot be calculated for small variables, but can be normally derived for the remaining ones. An example and further comments are given in section 3.2.

### 2.4. *Bodies and forces*

A rigid body is defined in *MBSymba*by specifying the reference frame to which the body is attached to, and by indicating its mass and inertia tensor with respect to the center of mass, which is assumed coincident with the frame origin. For example, the car chassis is defined as a body attached to the reference frame $\mathbf{T}_C$, with mass $m_c$ and the following inertia tensor:

$$\mathbf{I} = \begin{bmatrix} I_{c,xx} & 0 & -I_{c,xz} \\ 0 & I_{c,yy} & 0 \\ -I_{c,xz} & 0 & I_{c,zz} \end{bmatrix}_C$$

where the body symmetry with respect to the $xz$ plane is exploited at canceling the terms $I_{yz} = I_{xy} = 0$ at symbolic level. Similarly, the unsprung mass body is attached to the reference frame $\mathbf{T}_U$, has mass $m_u$ and the only nonzero element of the inertia tensor is $I_{u,zz}$. Finally, the trailer reference frame $\mathbf{T}_T$ is defined as:

$$\mathbf{T}_T = \begin{bmatrix} \cos\alpha & -\sin\alpha & \beta\cos\alpha & x_t \\ \sin\alpha & \cos\alpha & \beta\sin\alpha & y_t \\ -\beta & 0 & 1 & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

which is non-linear with respect to the trailer relative yaw $\alpha$ and linear with respect to the pitch $\beta$. The trailer body description is then completed by the mass $m_t$ and the principal moments of inertia $I_{t,xx}, I_{t,yy}, I_{t,zz}$. The next modelling step is the definition of active forces. In *MBSymba*, a force is defined by specifying its cartesian components with respect to a certain frame, together with the application point as well as the body to which the force is applied to. For example, to define the front right tyre force, see figure 1, the contact point need be defined first $\mathbf{C}_{R1} = (a_1, b_1, 0, 1)^T$ where $a_1$ is the front axle distance from the gravity centre $\mathbf{C}$ and $b_1$ is the front half track. Then the tyre force $\mathbf{F}_{R1} = (0, Y_{R1}, -N_{R1}, 0)_\delta^T$ is defined with respect to the following steered reference frame:

$$\mathbf{T}_\delta = \begin{bmatrix} \cos\delta & -\sin\delta & 0 & 0 \\ \sin\delta & \cos\delta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

Similarly, the front left tyre force is defined as a vector $\mathbf{F}_{L1} = (0, Y_{L1}, -N_{L1}, 0)_\delta^T$ applied to the point $\mathbf{C}_{L1} = (a_1, -b_1, 0, 1)^T$. Please note that there is no longitudinal tyre force, because the car is assumed rear-wheel-driven and the rolling resistance is neglected for simplicity. On the rear axle, the right tyre force is a vector $\mathbf{F}_{R2} = (X_2/2, Y_{R2}, -N_{R2}, 0)$ applied to $\mathbf{C}_{R2} = (-a_2, b_2, 0, 1)^T$, while the left one is a vector $\mathbf{F}_{L2} = (X_2/2, Y_{L2}, -N_{L2}, 0)$ applied to $\mathbf{C}_{L2} = (-a_2, -b_2, 0, 1)^T$. Since the car is fitted with an open differential, the longitudinal driving force $X_2$ is equally split between the two sides. Another external force acting on the chassis is the aerodynamics drag resistance $(-F_D, 0, 0, 0)$, which is applied to the chassis center $\mathbf{C}$.

On the trailer, the left and rear contact points are respectively $\mathbf{C}_{L3} = (-a_3 - r_3\beta, -b_3, h_t, 1)_T^T$ and $\mathbf{C}_{R3} = (-a_3 - r_3\beta, b_3, h_t, 1)_T^T$, where suffix $_T$ indicates that coordinates are given in the frame $\mathbf{T}_T$, equation (13). To define tyre forces, it is convenient to use reference frame which lies below the centre of gravity $\mathbf{G}_T$ at road level:

$$\mathbf{T}_\alpha = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & x_t \\ \sin\alpha & \cos\alpha & 0 & y_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

Now, left tyre force is a vector $\mathbf{F}_{L3} = (0, Y_{L3}, -N_{L3}, 0)_\alpha^T$ applied to $\mathbf{C}_{R3}$ and right force is a vector $\mathbf{F}_{R3} = (0, Y_{R3}, -N_{R3}, 0)_\alpha^T$ applied to $\mathbf{C}_{R3}$.

Finally, weight must be added to all bodies. For this purpose, it is sufficient to define the gravity acceleration field by setting the predefined variable gravity:

$$\mathbf{g} = \begin{pmatrix} 0 \\ 0 \\ g \\ 0 \end{pmatrix} \tag{16}$$

## 2.5. *Constraints*

The category of kinematic constraints includes a large variety of distinct cases. Moreover, a specific kinematic constraint may be defined using different yet equivalent sets of equations. These equations are typically non-linear, so the conversion from one set of constraint equations to an equivalent one may not be straightforward and sometimes not suitable for automatic manipulation. For such reasons, *MBSymba* leaves the user maximum freedom and flexibility to deal with kinematic constraints, which are simply defined with a list of implicit constraint equations and a list of generalized forces associated to each constraint equation.

In the current model, the tyre road contact conditions restrain two degrees of freedom of the trailer, namely the bounce and roll. However, since the roll constraint has been implicitly included by setting the trailer roll angle to zero, only the bounce constraint need be considered. The corresponding constraint equation is defined by setting the distance between the axle center $\mathbf{O}_3$ and the road plane to a constant value equal to the tyre radius, yielding to:

$$a_3\beta + h_t + z_t = 0 \tag{17}$$

where the generalized force $N_3$ associated to the constraint has the meaning of the total axle tyre load $N_3 = N_{R3} + N_{L3}$.

In addition to this general constraint definition just discussed, *MBSymba* provides a set of more structured procedures to create the most common constraints such as spherical, cylindrical, revolute, planar, prismatic and rod joints. For example, the hitch trailer that connects the trailer to the car is a *spherical* joint. To create this constraint, it is necessary first to create a point $\mathbf{H}_C = (-f_x, 0, f_z, 1)_C^T$ attached to the chassis and a point $\mathbf{H}_T = (j_x, 0, j_z, 1)_T^T$ attached to the trailer. At this points, *MBSymba* automatically calculates the constraint equations that correspond to the overlap of $\mathbf{H}_C$ and $\mathbf{H}_T$:

$$\begin{gathered} x_t + f_x + (j_z\beta + j_x)\cos\alpha - f_z\mu = 0 \\ y_t + (j_z\beta + j_x)\sin\alpha + f_z\phi = 0 \\ -(j_x\beta + f_x\mu + z) = 0 \end{gathered} \tag{18}$$

Concurrently, *MBSymba* creates the constraint force $\mathbf{H} = (H_x, H_y, H_z, 0)^T$ which acts on the chassis in $\mathbf{H}_C$ and reacts on the trailer in $\mathbf{H}_T$.

## 3. Equations of motion

Once the system has been described and a set of coordinates has been established, there are still many methods for deriving the equations of motion (Suh and Radcliffe (1978), Meirovitch (1970),Kane and Levinson (1985),Arnold (1989),Haug (1989),de Jalon and Bayo (1994),Torok (2000)), which range from direct application of Newton's laws to methods of analytical mechanics, like Lagrange's equations (which are available in *MBSymba*) as well as Kane's or Hamilton's equations (which are not available in *MBSymba*). Potentially, each of the above methods can be codified in a standard sequence of operations and the equations of motion derived in a completely automatic way. Different methods will lead to different set of equations, obviously equivalent to one another. However, the adoption of such algorithms would deprive the user of any possibility of inspecting and further manipulating the intermediate result, which may be very useful when developing models at symbolic level. Therefore *MBSymba* offers a set of powerful commands to deal with Newton's and Lagrange's dynamics, but without embedding them in a rigid, predefined framework. In this manner, the user has the possibility, and the responsibility, to select step by step the most profitable approach and to inspect and manipulate intermediate results. The advantages of this approach will be highlighted in the following sections: the first one illustrates the application of Newton's method by deriving the full set of equations of motion of the car-trailer system, while the second will discuss Lagrange's approach even though, as highlighted in section 2.3, it is not suitable to deal with systems that include infinitesimal variables.

### 3.1. *Newton's approach*

The equations of motion of a rigid body may be derived by means of the well known cardinal equations:

$$\frac{d}{dt}\mathbf{Q} = \mathbf{F} \tag{19}$$

$$\frac{d}{dt}\mathbf{K}_A + \mathbf{V}_A \times \mathbf{Q} = \mathbf{M}_A \tag{20}$$

Equation (19) describes the translation motion in terms of the linear momentum $\mathbf{Q}$ and the overall active and constraints forces $\mathbf{F}$ acting on the body, while equation (20) describes the rotation in terms of the angular momentum $\mathbf{K}$ and the moment of forces $\mathbf{F}$ with respect to the pole $\mathbf{A}$, which may move with velocity $\mathbf{V}_A$. These equations have been implemented in *MBSymba*, including the case of a non-inertial reference base frame, in which case the linear momentum and angular momentum are derived according to (6).

In a system composed of several bodies, cardinal equations (19, 20) may be written for each single body. Therefore, the number of equations depends on the number of bodies rather than the number of degrees of freedom, as the unknowns will include constraint forces too. However constraint forces may be partially or completely eliminated by further manipulating cardinal equations. In *MBSymba*, this task is totally left to the user, who has the responsibility of collecting a consistent set of equations, but also the maximum freedom to persue the most convenient strategy to do so. *MBSymba* also gives the possibility of calculating cardinal equations for a set of two or more bodies, which has the advantage of automatically eliminating the internal reaction forces.

In this case study, the behaviour of the car-only and the car-trailer system will be compared, so it is convenient to derive separate equations for the two subsystems. The car includes two rigid bodies (chassis and unsprung mass). *MBSymba* allows us to calculate the cardinal equations for the whole system without caring about the internal reaction forces. To define the *car* subsystem it is sufficient to collect bodies, active forces (air drag, tyre forces) and reactive forces (hook) into a set and then *MBSymba* automatically derive Newton's equations (19). The result is a vector that can be projected onto any reference frame, in particular the components along the moving frame $x, y, z$ axes read:

$$-(m_u + m_c)vr = X_2 - (Y_{L1} + Y_{R1})\sin\delta + H_x \tag{21a}$$

$$(m_u + m_c)(ur + \dot{v}) = (Y_{R1} + Y_{L1})\cos\delta + (Y_{L2} + Y_{R2}) + H_y \tag{21b}$$

$$m_c\ddot{z} = (m_u + m_c)g - (N_{L1} + N_{L2} + N_{R1} + N_{R2}) + H_z \tag{21c}$$

By selecting the origin of the moving frame as a pole, Euler's equations (20) becomes:

$$I_{xx}\ddot{\phi} - I_{c,xz}\dot{r} + (I_{c,zz} - I_{c,yy} - I_{c,xz})r\dot{\mu} + (I_{c,zz} - I_{c,yy})\phi r^2 + \\ +((h_c - z)m_c + h_u m_u)ur + (h_c m_c + h_u m_u)\dot{v} = M_{A,x} \tag{22a}$$

$$I_{c,yy}\ddot{\mu} + (I_{c,xx} + I_{c,yy} - I_{c,zz})r\dot{\phi} + ((-I_{c,xx} + I_{c,zz})\mu - I_{c,xz})r^2 + \\ +(h_c m_c + h_u m_u)vr = M_{A,y} \tag{22b}$$

$$(I_{c,zz} + I_{u,zz})\dot{r} - I_{c,xz}\ddot{\phi} + 2I_{c,xz}r\dot{\mu} = M_{A,z} \tag{22c}$$

where

$$M_{A,x} = b_1(N_{L1} - N_{R1}) + b_2(N_{L2} - N_{R2}) - (\mu f_x + f_z - h_c + z)H_y - f_z\phi H_z$$
$$M_{A,y} = a_1(N_{R1} - N_{L1}) + a_2(N_{R2} - N_{L2}) + (\mu f_x + f_z - h_c + z)H_x + (f_x - f_z\mu)H_z$$
$$M_{A,z} = a_1(Y_{L1} + Y_{R1})\cos\delta - b_1(Y_{L1} - Y_{R1})\sin\delta - a_2(Y_{L2} + Y_{R2}) + f_z\phi H_x + (f_z\mu - f_x)H_y$$

It is worth emphasising Euler's equations may be calculated with respect to any pole chosen by the user. In this specific case the selection of the origin of the moving frame leads to quite simple expressions for $M_{A,x}, M_{A,y}$ that do not contain any lateral and longitudinal tyre forces. This is an example of the flexibility of *MBSymba*, where the user retains the freedom to select the most convenient equations.

The trailer assembly includes the trailer rigid body, tyre forces and hook reaction force, the corresponding Newton-Euler's equations are:

$$\begin{aligned} m_t\left(\ddot{x}_t - vr - \dot{r}y_t - r^2 x_t - 2r\dot{y}_t\right) &= -(Y_{L3} + Y_{R3})\sin\alpha - H_x \\ m_t\left(\ddot{y}_t + ur + \dot{v} + \dot{r}x_t - r^2 y_t + 2r\dot{x}_t\right) &= (Y_{L3} + Y_{R3})\cos\alpha - H_y \\ m_t\ddot{z}_t &= m_t g - (N_{L3} + N_{R3}) - H_z \end{aligned} \tag{23}$$

$$(I_{t,zz} - I_{t,xx} - I_{t,yy})r\dot{\beta} = M_{T,x} \tag{24a}$$

$$I_{t,yy}\ddot{\beta} + (I_{t,zz} - I_{t,xx})r^2\beta = M_{T,y} \tag{24b}$$

$$I_{t,zz}(\dot{r} + \ddot{\alpha}) = M_{T,z} \tag{24c}$$

where

$$M_{T,x} = (j_x\beta - j_z)(H_x\sin\alpha - H_y\cos\alpha) - (h_t + a_3\beta)(Y_{L3} + Y_{R3}) + b_3(N_{L3} - N_{R3})$$
$$M_{T,y} = (j_x\beta - j_z)(H_x\cos\alpha + H_y\sin\alpha) + [(h_t - r_3)\beta - a_3](N_{L3} + N_{R3}) + (j_x + j_z\beta)H_z$$
$$M_{T,z} = (j_x + j_z\beta)(H_x\sin\alpha - H_y\cos\alpha) + [(h_t - r_3)\beta - a_3](Y_{L3} + Y_{R3})$$

At this point, the car and trailer mathematical model consists of 16 equations: twelve Newton-Euler equations (21-24), eleven of which are second order differential equations while (21a) is algebraic because the forward speed $u$ is assumed constant, plus four algebraic constraints (17-18). The list of unknowns includes 20 elements: the gross motion velocity variables $v, r$, chassis motion $z, \phi, \mu$, the longitudinal tyre force $X_2$, trailer coordinates $x_t, y_t, z_t, \alpha, \beta$, trailer hitch forces $H_x, H_y, H_z$, and tyre loads $N_{R1}, N_{L1}, N_{R2}, N_{L2}, N_{R3}, N_{L3}$. The four remaining equations are those relating the normal loads of car tyres to the sprung motion, which will be discussed in the tyre section.

### 3.2. *Lagrange's approach*

For a multibody system described by means of a set on $n$ dependent variables $\mathbf{q} = \{q_1, q_2 \dots q_N\}^T$ and subject to $m < n$ algebraic constraint equations:

$$\varphi_i(\mathbf{q}, t) = 0, \qquad i = 1, 2 \dots m \tag{25}$$

Lagrange's equations of motion are:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i - \sum_{k=1}^{m} \lambda_k \frac{\partial \varphi_k}{\partial q_i}, \qquad i = 1 \dots n \tag{26}$$

where $L = T - V$ is the lagrangian, $T$ is the kinetic energy, $V$ is the potential energy, $Q_i$ are generalized forces that may be calculated from the application of the virtual work principle and $\lambda_j$ are Lagrange's multipliers. Lagrange's approach is more popular than Newton's one mainly because Lagrange's equations do not contain reaction forces (leading to a smaller number of equations than Newton's approach). As an example, Lagrange's equation with respect to the trailer relative yaw, i.e. with respect to the non-inifinitesimal variable $\alpha$, may be derived as follows:

$$I_{t,zz}(\dot{r} + \ddot{\alpha}) = (j_x + j_z\beta)(H_x \sin\alpha - H_y \cos\alpha) + [(h_t - r_3)\beta - a_3](Y_{L3} + Y_{R3}) \tag{27}$$

It may be observed that equation (27) is identical to (24c) that was derived using Newton's approach. However in (27) variables $H_x, H_y$ are actually Lagrange's multipliers, while in (24c) they represents the constraint force $\mathbf{H}$. Such formal and substantial equivalence is obtained because *MBSymba* uses a proper scaling while creating constraints such as the spherical joint (18).

The problem with Lagrange's approach is that it is not compatible with the linear modelling option offered by *MBSymba*. Indeed, as anticipated is section 2.3, in order to preserve all first order infinitesimal terms in the equations of motion, both kinetic energy and virtual works calculation need be computed by including also second order terms. Therefore, if one attempts to derive Lagrange's equation with respect to the trailer infinitesimal pitch angle would obtain:

$$I_{t,yy}\ddot{\beta} - I_{t,xx}r^2\beta = j_z(H_x \cos\alpha + H_y \sin\alpha) + j_x H_z - (a_3 + r_3\beta)(N_{L3} + N_{R3})$$

This equation is wrong because it is missing some inertial terms as well as some force terms, which were preserved in Newton's equations of motion (24b). If one wants to use Lagrange's approach, all non-linear kinematic terms along the whole modelling process should be retained, the non-linear Lagrange equations derived, and finally linearized. In most cases this in not convenient as the advantages of dealing with simpler expressions from the beginning overcomes the burden of having to define constraint forces.

Another issue with Lagrange's equations arises when the model is developed in a non-inertial reference frame by using quasi coordinates such as $v$ and $r$. Since quasi coordinates describes the gross motion in term of velocity instead of position, Lagrange's equations (26) may not be used in their original form but, as explained in Meirovitch (1970); Pacejka (2006), should be rearranged as follows:

$$\frac{d}{dt}\frac{\partial T}{\partial v} + r\frac{\partial T}{\partial u} = Q_v \tag{28a}$$

$$\frac{d}{dt}\frac{\partial T}{\partial r} - v\frac{\partial T}{\partial u} + u\frac{\partial T}{\partial v} = Q_r \tag{28b}$$

Unfortunately, if the lateral velocity coordinate $v$ is replaced, for instance, by the vehicle sideslip (i.e. the angle between forward and lateral velocity) another formulation has to be used. If a moving frame with spatial motion is used, for example to model an aircraft, Lagrange's equations will be different again and more complicated. In *MBSymba*, these issues may be completely avoided by using, again, Newton's approach. In particular, if Newton's equations are calculated for the *whole* system, it is not necessary to define constraint forces because they are internal to the system and will not appear in the resulting equations in any case. For example, the $y$ component of Newton's translation equation for the whole system is

$$\begin{aligned}(m_u + m_c + m_t)\,(ur + \dot{v}) + m_t\left(\ddot{y}_t + \dot{r}x_t + 2r\dot{x}_t - r^2 y_t\right) = \\ = (Y_{R1} + Y_{L1})\cos\delta + (Y_{L2} + Y_{R2}) + (Y_{L3} + Y_{R3})\cos\alpha\end{aligned} \tag{29}$$

and actually corresponds to the Lagrange equation (28a). Since *MBSymba* offers the possibility of mixing both Lagrange's and Newton's approaches, the presence of constraint forces may be avoided by using Newton's equations to deal with quasi-coordinates and Lagrange's equations for the other coordinates.

In conclusion, *MBSymba* offers all the tools necessary to derive the Lagrange equations of a multibody system defined either in a inertial or non inertial base, either unconstrained or constrained. However, if some variables have been declared infinitesimal, Lagrange's equation with respect to such variables cannot be calculated properly.

### 3.3. *Tyre and suspension models*

The Newton-Euler equations (21)-(24) include tyre vertical, longitudinal and lateral forces which will now be defined. Tyres are assumed to be rigid, while suspensions are modelled as linear spring-damper elements that move vertically. Therefore, the vertical load on each tyre may be calculated as the sum of the wheel weight and the suspension force as follows:

$$N_{R1,L1} = \frac{a_2}{2w}(m_c + m_u)g + k_1(z - a_1\mu \pm b_1\phi) + c_1(\dot{z} - a_1\dot{\mu} \pm b_1\dot{\phi}) \tag{30a}$$

$$N_{R2,L2} = \frac{a_1}{2w}(m_c + m_u)g + k_2(z + a_2\mu \pm b_2\phi) + c_1(\dot{z} + a_2\dot{\mu} \pm b_2\dot{\phi}) \tag{30b}$$

where $k_1, k_2$ are the front and rear suspension stiffness, $c_1, c_2$ are the damping coefficients, and finally sign $+$ is for the right side and $-$ for the left side. In steady state conditions, tyre lateral forces mainly depend on the sideslip angle $\lambda$, i.e. the angle between the direction of travel and the equatorial plane of the tyre (as depicted in figure 2) and the normal load $N$. Quite often tyre forces are calculated according to Pacejka's magic formula Pacejka (2006), which is exhaustive but quite complex. For the sake of simplicity the following simplified form, which still preserves the non-linearity with respect to the sideslip angle $\lambda$ and normal load $N$ is adopted herein:

$$Y_0(\lambda, N) = D_y \sin\left[B_y \arctan(C_y\lambda)\right]\left[1 + \alpha\left(1 - \tfrac{N}{N_0}\right)\right] N \tag{31}$$

As depicted in figure 2, this expression still captures the force non-linear dependency on the tyre load $N$ as well as the sideslip angle $\lambda$. The latter may be calculated as follows:

$$\lambda = -\arctan\frac{V_{C,y}}{V_{C,x}} \tag{32}$$

where $V_{C,x}$ and $V_{C,y}$ are the longitudinal and lateral component of the the contact point velocity respectively, as shown in Figure 2. For the different tyres, one obtains:

$$\begin{aligned} V_{C1,x} &= \quad (u \pm b_1 r)\cos\delta + (v + a_1 r)\sin\delta \\ V_{C1,y} &= -(u \pm b_1 r)\sin\delta + (v + a_1 r)\cos\delta \end{aligned} \tag{33a}$$

$$\begin{aligned} V_{C2,x} &= u \pm b_2 r \\ V_{C2,y} &= v - a_2 r \end{aligned} \tag{33b}$$

$$\begin{aligned} V_{C3,x} &= (v + x_t r + \dot{y}_t)\sin\alpha + (u + \dot{x}_t - y_t r)\cos\alpha + (h_t - r_3)\dot{\beta} \pm b_3(r + \dot{\alpha}) \\ V_{C3,y} &= (v + x_t r + \dot{y}_t)\cos\alpha - (u + \dot{x}_t - y_t r)\sin\alpha + (h_t - r_3)\beta r \end{aligned} \tag{33c}$$

Where sign $+$ applies to the left tyres while sign $-$ applies to the right ones. Equation (31) is valid in steady state conditions, while transient tyre forces may be more accurately described by a first order relaxation equation Pacejka (2006) as follows:

$$\frac{\sigma}{V_{C,x}}\dot{Y} + Y = Y_0\left(-\arctan\frac{V_{C,y}}{V_{C,x}}, N\right) \tag{34}$$

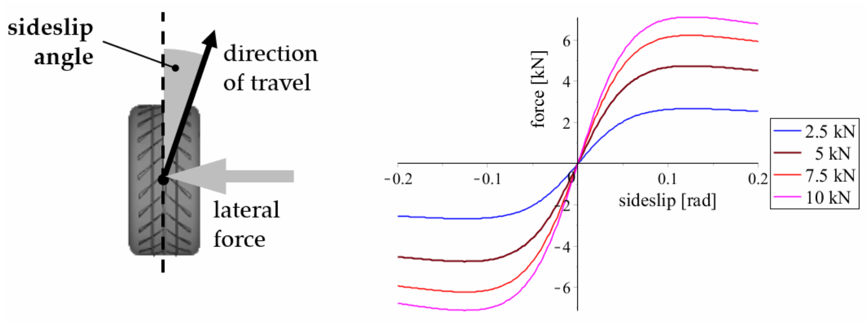


Fig. 2. Lateral force.

In order to avoid the numerical singularity at zero speed, the above equations may be rewritten as:

$$\sigma \dot{Y} = V_{C,x} \left[ Y_0 \left( -\arctan \frac{V_{C,y}}{V_{C,x}}, N \right) - Y \right] \tag{35}$$

It is worth pointing out that the calculation of tyre forces just described could be easily combined into a Maple procedure. In general, the user may create new objects on the top of *MBSymba* by specifying rules for type definition and manipulation.

## 4. State space formulation

The following two sections explain how to collect and convert the equations of motion into a state space formulation, which is the most suitable for time integration, stability analysis, and control design. In particular, it is shown how the equations of motion of the car (without the trailer) may be easily converted into a non-linear state space formulation $\dot{\mathbf{x}}_c = \mathbf{F}(\mathbf{x}_c, \mathbf{u})$. However, the car with trailer model includes some differential-algebraic equations (DAE) and yields to an implicit state space formulation $\mathbf{A}(\mathbf{x})\dot{\mathbf{x}} = \mathbf{B}(\mathbf{x}, \mathbf{u})$ where $\mathbf{A}$ is not invertible.

### 4.1. *Car model*

The car model consists of two rigid bodies, unsprung mass and chassis. Since the forward speed $u$ is constant, the model has five degrees of freedom only: the gross motion is defined in term of lateral speed $v$ and yaw rate $r$, while the chassis motion (due to suspensions) is described in terms of bounce $z$, roll $\phi$ and pitch $\mu$. There are six Newton-Euler equations (21),(22), but (21a) is algebraic because $u$ is constant. This equation may be discarded because it is not coupled with the others. In the remaining equations, the chassis acceleration contains second order terms $\ddot{z}, \ddot{\phi}, \ddot{\mu}$, while the gross motion contains first order terms $\dot{v}, \dot{r}$ only. For this reason, a first order state-space formulation is here preferred instead of the more commonly adopted second order formulation, e.g. de Jalon and Bayo (1994). The reduction of Newton-Euler equations to the first order is obtained by introducing in (21),(22) the following velocity variables:

$$\begin{aligned} V_z &= \dot{z} \\ \Omega_\phi &= \dot{\phi} \\ \Omega_\mu &= \dot{\mu} \end{aligned} \tag{36}$$

To isolate the car, it is necessary to unlink the trailer by setting $H_x = H_y = H_z = 0$. It is also necessary to consider tyre force expressions (30)-(33) and to include four first order tyre relaxation equations (35) (one for each tyre). When collecting equations one obtains the following non-linear, implicit state space formulation:

$$\mathbf{A}_c \dot{\mathbf{x}}_c = \mathbf{B}(\mathbf{x}_c, \mathbf{u}) \tag{37}$$

where $\mathbf{x}_c$ is the vector of the twelve *independent* state variables:

$$\mathbf{x}_c = \{z, \mu, \phi, v, r, V_z, \Omega_\mu, \Omega_\phi, Y_{R1}, Y_{L1}, Y_{R2}, Y_{L2}\}^T \tag{38}$$

and $\mathbf{u}$ is the vector of inputs, which in this case consists of the sole steering angle:

$$\mathbf{u} = \{\delta\} \tag{39}$$

The state space matrix $\mathbf{A}_c$ and the vector $\mathbf{B}_c$ are not reported here for brevity. However, in this case the matrix $\mathbf{A}_c$ is constant because it depends only on vehicle inertia and tyre relaxation, and moreover it is invertible because of the independent coordinate formulations. By symbolically calculating $\mathbf{A}_c^{-1}$, equation (37) may be rewritten in the following explicit, non-linear, state space formulation:

$$\dot{\mathbf{x}}_\mathbf{c} = \mathbf{A}_c^{-1}\mathbf{B}(\mathbf{x}_c, \mathbf{u}) = \mathbf{F}(\mathbf{x}_c, \mathbf{u}) \tag{40}$$

which is the most suitable for time integration, stability analysis, and control design.

### 4.2. *Car and trailer model*

The state space equations of the full model can be assembled in a fashion similar to what was done with the sole car. The first step is the reduction to the first order of the Newton-Euler equations (23),(24) by the introduction of the following velocity variables:

$$\begin{aligned} V_{xt} &= \dot{x}_t \\ V_{yt} &= \dot{y}_t \\ V_{zt} &= \dot{z}_t \\ \Omega_\alpha &= \dot{\alpha} \\ \Omega_\beta &= \dot{\beta} \end{aligned} \tag{41}$$

Again, it is necessary to consider tyre force expressions as in section 3.3 and to include two first order tyre relaxation equations (34) for the trailer wheels. In conclusion, the full set of implicit state space equations (42) includes five car dynamics equations (21b),(21c),(22), six trailer dynamics equations (23),(24), eight velocity equations (36)-(41), six tyre relaxation equations (34) and four constraints equations (17),(18), for a total of $n = 29$ equations and as much state variables as follows:

$$\mathbf{A}(\mathbf{x})\dot{\mathbf{x}} = \mathbf{B}(\mathbf{x}, \mathbf{u}) \tag{42}$$

where $\mathbf{x} = \mathbf{x}_c \cup \mathbf{x}_t$, with $\mathbf{x}_c$ from (38) and $\mathbf{x}_t$ the additional trailer state variables:

$$\mathbf{x}_t = \{x_t, y_t, z_t, \alpha, \beta, V_{xt}, V_{yt}, V_{zt}, \Omega_\alpha, \Omega_\beta, N_{R3}, N_{L3}, Y_{R3}, Y_{L3}, H_x, H_y, H_z\}^T \tag{43}$$

Because of the algebraic constraints, the matrix $\mathbf{A}(\mathbf{x})$ is singular and cannot be inverted. More precisely, (42) is a set of differential-algebraic equations (DAE) of order three, that can be integrated by using a solver like DASSL, MEBDFI or PSIDE (Test set for IVP solvers, Kunkel and Mehrmann (2006)). However, it is worth pointing out that the equations of motion may be further manipulated and converted into a different formulation. Among many possibilities, it is possible for example to reduce the DAE index of equations (42) by stabilising constraints equations (17),(18) as described in Baumgarte (1972); de Jalon and Bayo (1994). In particular, the algebraic constraint (17) may be easily converted into a differential one and stabilised as follows:

$$\left(a_3\ddot{\beta} + \ddot{z}_t\right) + 2\zeta\omega\left(a_3\dot{\beta} + \dot{z}_t\right) + \omega^2\left(a_3\beta + z_t + h_t\right) \tag{44}$$

and the same method may be used for equation (18) too.

## 5. Simulations

Some numerical results are now given to demonstrate the effectiveness of the proposed approach, which conveniently mixes infinitesimal and non-infinitesimal variables.

### 5.1. *Open loop simulations in time domain*

State space equations (37) and (42) can be conveniently converted to Matlab, Fortran or C code using the built-in Maple export command and then embedded into a simulation program. However, since the problem size is relatively small, in this case simulations may be carried out in Maple. Indeed, the car equations (40) are formulated in ODE form and may be integrated in Maple without further manipulation. As an example, figure 3 shows the vehicle response to a steering angle linear ramp which (saturated at 0.1 rad after 5 s) in terms of the yaw rate $r$ and lateral speed $v$. The figure on the left has been obtained at a speed of 10 m/s: the response trend basically corresponds to a sequence of steady state conditions, where the lateral acceleration varies from 0 to approximately 3 m/s$^2$. Positive $v$ means that the car is travelling 'nose-out'. The figure on the right has been obtained at a speed of 20 m/s and is dramatically different. Indeed a factor of two increase in the speed would result in a factor four increase of the lateral acceleration, which far exceeds tyre adherence limits. For such reasons the yaw rate amplitude remains small while large oscillations are evident both on the yaw rate and on the lateral speed $v$ (which is now negative, indicating that the car is travelling 'nose-in'). Tyre lateral forces are shown in figure 4: at a low speed, their regular, practically monotone, variation with time makes it possible to appreciate that tyre forces are greater on the outer side (i.e. the left one), as it is obviously expected for a car travelling on a clockwise turn. At high speed, it is clearly visible that tyre forces oscillations are out of phase between the front and rear axle, and therefore are responsible for the yaw rate oscillations in figure 3. In addition, tyres reach their adherence limits in the time interval $3 < t < 4$, which corresponds to the beginning of the yaw and lateral oscillations, which are only slightly damped as they are still present after more than 10 s. During these oscillations, the chassis motion remains compatible with the assumption of infinitesimal motion. This example illustrates the convenience of

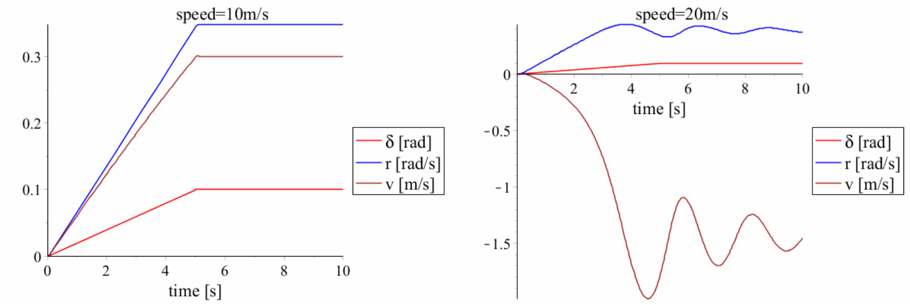


Fig. 3. Yaw rate $r$ and lateral speed $v$ response to a steering angle ramp input at different speeds.

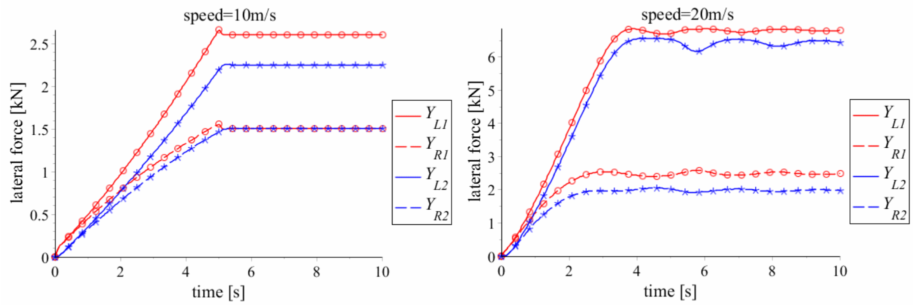


Fig. 4. Tyre lateral force response to a steering angle ramp input at different speeds.

modelling system non-linearities only when they are significant, e.g. on tyre force calculation.

## 5.2. *Steady cornering*

The stationary manoeuver at constant speed and constant yaw rate is widely used to assess basic cornering performance of ground vehicles Pacejka (2006). From a mathematical point of view, stationary equations of motion are obtained from the full equations (42) by setting both the state variables and the inputs to a constant value, i.e. $\mathbf{x}(t) = \mathbf{x}_0$ and $\delta(t) = \delta_0$, yielding to the following set of non-linear algebraic equations:

$$\mathbf{B}(\mathbf{x}_0, \delta_0) = \mathbf{0} \tag{45}$$

where the stationary state variables $\mathbf{x}_0$ may be calculated for each given value of the steering angle $\delta_0$. It is worth pointing out that stationary conditions $\mathbf{x}_0$ exist only because a proper set of state variables that includes moving frame velocities $u, v$ and exclude any fixed frame coordinate (such as vehicle trajectory $x, y$) have been explicitly selected during the modelling phase, thanks to the flexibility offered by *MBSymba* in the choice of coordinates. Simulation results that compare the behaviour of the car with and without the trailer are shown in Figures 5 and 6. In more detail, Figure 5a shows tyre vertical load for all car corners varying as a

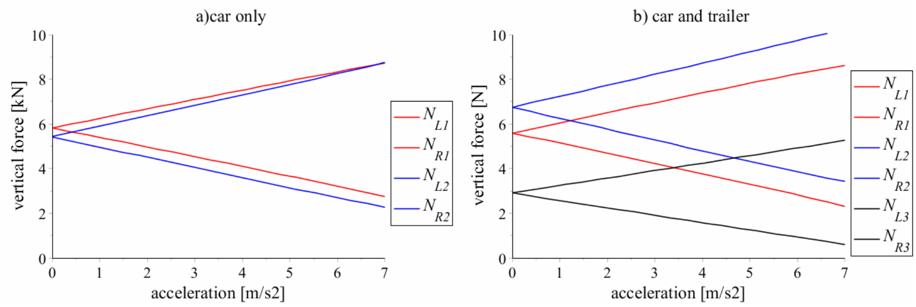


Fig. 5. Tyre vertical loads as a function of the lateral acceleration, cornering radius R=100m.

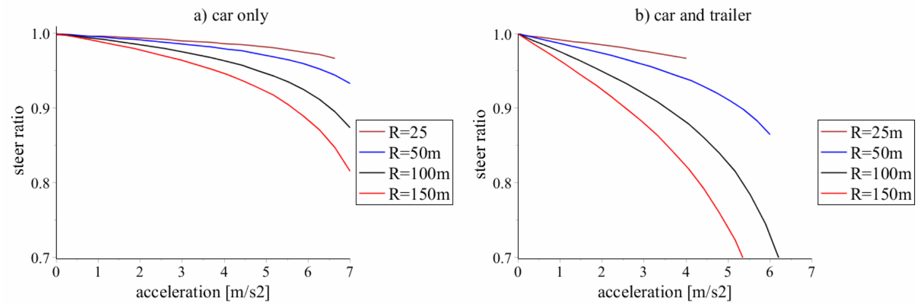


Fig. 6. Steering ratio as a function of the lateral acceleration, different cornering radii.

function of the lateral acceleration, on a constant radius trajectory of $R = 100$ m. There, centrifugal force obviously creates a lateral load transfer from the inner (right) side tyres to the outer (left) side. It is also evident that, in straight running (zero lateral acceleration) the load on the rear axle is smaller than the load on the front, as expected because the center of gravity is closer to the front axle. However, as the lateral acceleration increases, the inner/outer load transfer on the rear axle is greater than that of the front axle because the rear suspension is stiffer than the front suspension. Figure 5a shows that when the trailer is attached to the car two main differences appear. The load on the rear axle significantly increases, because it supports a portion of the trailer weight, and the trailer inner tyre load decreases very quickly because of an unfavourable height to width ratio $h/b$ which makes the trailer at risk of rollover at high lateral accelerations. Lateral load transfer has also a significant influence on steering performance, which may be assessed by analysing the ratio between the actual cornering radius $R$ and Ackermann's radius $R_a$[a]

$$\sigma = \frac{R}{R_a} = \frac{R \tan \delta}{a_1 + a_2} \tag{46}$$

as depicted in figure 6.

[a] $R_a$ is the cornering radius calculated by neglecting tyre sideslips.

Table 1. Model parameters.

| parameter | value | | description |
|---|---|---|---|
| $g$ | 9.806 | $m/s^2$ | gravity acceleration |
| $m_u$ | 300 | kg | car unsprung mass |
| $h_u$ | 0.35 | m | car unsprung mass height |
| $m_c$ | 2400 | kg | car chassis mass (sprung) |
| $h_c$ | 0.75 | m | car chassis centre of mass height |
| $a_1$ | 1.40 | m | distance between the front axle and the vehicle centre of mass |
| $a_2$ | 1.50 | m | distance from the rear axle and the vehicle centre of mass |
| $a_1 + a_2$ | 2.67 | m | wheelbase |
| $I_{u,zz}$ | 560 | $kg\,m^2$ | unsprung mass, yaw moment of inertia |
| $I_{c,xx}$ | 2000 | $kg\,m^2$ | chassis, roll moment of inertia |
| $I_{c,yy}$ | 2800 | $kg\,m^2$ | chassis, pitch moment of inertia |
| $I_{c,zz}$ | 3800 | $kg\,m^2$ | chassis, yaw moment of inertia |
| $I_{c,xz}$ | 0 | $kg\,m^2$ | chassis, cross moment of inertia |
| $b_1$ | 0.78 | m | front axle half track |
| $b_2$ | 0.78 | m | rear axle half track |
| $k_1$ | 60 | kN/m | front suspension vertical stiffness |
| $c_1$ | 4 | kNs/m | front suspension vertical damping |
| $k_2$ | 65 | kN/m | rear suspension vertical stiffness |
| $c_2$ | 5 | kNs/m | rear suspension vertical damping |
| | | | |
| $m_t$ | 800 | kg | trailer mass (unsprung) |
| $a_3$ | 0.30 | m | distance between the wheels axle and trailer centre of mass trailer |
| $h_t$ | 0.65 | m | chassis centre of mass height |
| $I_{t,xx}$ | 200 | $kg\,m^2$ | trailer, roll moment of inertia |
| $I_{t,yy}$ | 250 | $kg\,m^2$ | trailer, pitch moment of inertia |
| $I_{t,zz}$ | 300 | $kg\,m^2$ | trailer, yaw moment of inertia |
| $I_{t,xz}$ | | $kg\,m^2$ | trailer, cross moment of inertia |
| $r_3$ | 0.30 | m | trailer wheels rolling radius |
| $b_3$ | 0.65 | m | trailer half track |
| $h_h$ | 0.4 | m | trailer hitch height |
| $f_x$ | 2.10 | m | distance between the trailer hitch and the vehicle centre of mass |
| $j_x$ | 0.80 | m | distance between the trailer hitch and the trailer centre of mass |
| | | | |
| $\sigma_1$ | 0.2 | m | front tyre relaxation length |
| $B_{y1}$ | 12 | | front tyre stiffness factor |
| $C_{y1}$ | 1.6 | | front tyre shape factor |
| $D_{y1}$ | 0.95 | | front tyre peak factor |
| $\alpha_1$ | 0.25 | | front tyre load dependency factor |
| $\sigma_2$ | 0.2 | m | rear tyre relaxation length |
| $B_{y2}$ | 12 | | rear tyre stiffness factor |
| $C_{y2}$ | 1.6 | | rear tyre shape factor |
| $D_{y2}$ | 0.90 | | rear tyre peak factor |
| $\alpha_2$ | 0.25 | | rear tyre load dependency factor |
| $\sigma_3$ | 0.2 | m | trailer tyre relaxation length |
| $B_{y3}$ | 12 | | trailer tyre stiffness factor |
| $C_{y3}$ | 1.6 | | trailer tyre shape factor |
| $D_{y3}$ | 0.95 | | trailer tyre peak factor |
| $\alpha_3$ | 0.25 | | trailer tyre load dependency factor |

When the lateral acceleration is small, the steering angle may be calculated according to kinematic considerations only, and the steering ratio factor is $\sigma = 1$, as confirmed in the figure. As the lateral acceleration increases, the steering ratio decreases and, for each constant cornering radius, the steering angle decreases as is typical of *oversteering* vehicles. Figure 6 highlights that, due to tyre non-linear behaviour, oversteering depends on cornering radius too; moreover the trailer dramatically increases the oversteering tendency.

### 5.3. *Stability*

In order to analyse stability in straight running and while cornering, model equations (42) are linearized as follows:

$$\mathbf{A}(\mathbf{x}_0)\dot{\mathbf{x}} = \frac{\partial \mathbf{B}}{\partial \mathbf{x}}(\mathbf{x} - \mathbf{x}_0) + \frac{\partial \mathbf{B}}{\partial \mathbf{u}}(\mathbf{u} - \mathbf{u}_0) \tag{47}$$

or, in a more compact form:

$$\mathbf{A}_0\dot{\mathbf{x}} = \mathbf{B}_{\mathbf{x}0}(\mathbf{x} - \mathbf{x}_0) + \mathbf{B}_{\mathbf{u}0}(\mathbf{u} - \mathbf{u}_0) \tag{48}$$

where $\mathbf{x}_0, \mathbf{u}_0$ is a solution of steady state equations (45). In multibody software, linearisation is usually carried by numerical differentiation, but this operation is characterised by the well known cancellation/truncation dilemma. To avoid this potential pitfall, *MBSymba* exploits Maple capabilities to linearize the equations of motion *symbolically*, therefore without introducing any numerical cancellation/truncation. Finally, vehicle stability may be assessed by converting equation (48) into the following generalized eigenvalue problem:

$$\mathbf{A}_0\lambda\mathbf{X} = \mathbf{B}_{\mathbf{x}0}\mathbf{X} \tag{49}$$

For the car alone, the matrix $\mathbf{A}_c$ has full rank hence the solution of problem (49) does not present difficulties. On the contrary, the car-trailer equations of motion (42) are differential-algebraic and therefore matrix $\mathbf{A}_0$ is singular. To avoid the presence of infinite eigenvalues that could irremediably hinder the accuracy of other eigenvalues, it is convenient eliminate algebraic equations before eigenvalues calculation. Reduction of the linear DAE equations (48) to an ODE system is much simpler than reducing non-linear DAE equations. For instance, a suitable algorithm based on matrix triangularization is described in Cossalter *et al.* (2011a).

## 6. Conclusions

The package *MBSymba* for the symbolic modelling of multibody systems has been described and, as an application example, a model of a car with an attached trailer has been presented. *MBSymba* includes a set of commands to conveniently define and manipulate multibody objects such as points, vectors, bodies, forces, torques and constraints. Other special objects, e.g. customized tyre models, may be additionally defined by the user.

The first advantage of *MBSymba* is the possibility of choosing the most suitable set of modelling variables, which in this case allowed for the utilization of quasi-coordinates in a moving frame approach, which is very appealing for the modelling

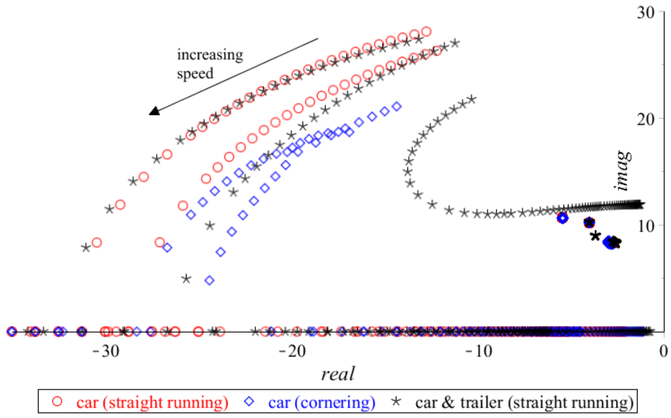


Fig. 7. Eigenvalues in the speed range from 5 to 50 m/s.

of ground, water or air vehicles. Another advantage of *MBSymba* is the capability of modelling systems by using a desired mix of infinitesimal and non-infinitesimal variables, with *MBSymba* automatically linearizing all relevant expressions as they arise, in a smart and optimized way. *MBSymba* also includes a set of procedures for the (quasi) automatic derivation of the equations of motion, including both Newton's cardinal equations and Lagrange's equations. Differences between the two approaches have been discussed with a focus on the presence of infinitesimal variables, highlighting that the step-by-step linearisation modelling technique is compatible with Newton's approach only.

Finally, examples of open loop time simulations, steady state and stability analysis have been included. These simulations may be performed either by exploiting Maple's capabilities, or by exporting the equations of motion to Matlab, Fortran, C/C++ or other coding languages to compile high performance simulation programs. *MBSymba* is freely available for education and research purposes at the `www.multibody.net/mbsymba/` website, which also includes tutorials and examples.

## References

Arnold, V. [1989] *Mathematical Methods of Classical Mechanics* (Springer-Verlag, New York).

Baumgarte, J. [1972] Numerical stabilization of the differential equations of keplerian motion, *Celestial Mechanics* **5**(4), 490–501, doi:10.1007/BF01464775.

Biral, F., da Lio, M., Lot, R. and Sartori, R. [2010] An intelligent curve warning system for powered two wheel vehicles, *European Transport Research Review* **2**(3), 147–156, doi:10.1007/s12544-010-0033-2.

Cossalter, V., Lot, R. and Massaro, M. [2011a] An advanced multibody code for handling and stability analysis of motorcycles, *Meccanica* **46**(5), 943–958.

Cossalter, V., Lot, R., Massaro, M. and Sartori, R. [2011b] Development and validation of an advanced motorcycle riding simulator, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* **225**(6), 705–720.

de Jalon, J. and Bayo, E. [1994] *inematic and Dynamic Simulation of Multibody Systems* (Springer Verlag, Berlin).

Docquier, N., Poncelet, A. and Fisette, P. [2013] Robotran: A powerful symbolic gnerator of multibody models, *Mechanical Sciences* **4**(1), 199–219, doi:10.5194/ms-4-199-2013.

DST (2016) [2016] URL `www.simwise4d.com`.

Goldstein, H., Poole, C. and Safko, J. [2000] *Classical Mechanics* (Addision-Wesley, New York).

Haug, E. [1989] *Computer-Aided Kinematics and Dynamics of Mechanical Systems* (Allyn and Bacon, Boston).

Kane, T. and Levinson, D. [1985] *Dynamics: Theory and Application* (McGraw-Hill, New York).

Kreuzer, E. [1979] *Symbolische Berechnung der Bewegungsgleichungen von Mehrkorpersystemen*, PhD thesis, VDI Fortschritt-Berichte, Reihe 11, Nr. 32. VDI-Verlag, Dusseldorf.

Kunkel, P. and Mehrmann, V. [2006] *Differential-algebraic equations: analysis and numerical solution.* (European Mathematical Society, Zrich), ISBN 978-3-03719-017-3.

Kurz, T., Eberhard, P., Henninger, C. and Schiehlen, W. [2010] From neweul to neweul-m2: Symbolical equations of motion for multibody system analysis and synthesis, *Multibody System Dynamics* **24**(1), 25–41, doi:10.1007/s11044-010-9187-x.

Levinson, D. [1977] Equations of motion for multi-rigid-body systems via symbolic manipulations, *Journal of Spacecraft Rockets* **14**.

Li, Z. and Kota, S. [2001] Virtual prototyping and motion simulation with adams, *Journal of Computing and Information Science in Engineering* , 276–279.

Lot, R. and Bianco, D. [2015] Lap time optimisation of a racing go-kart, *Vehicle System Dynamics* **51**(1-21), 1–10, doi:10.1080/00423114.2015.1125514.

Lot, R. and Da Lio, M. [2004] A symbolic approach for automatic generation of the equations of motion of multibody systems, *Multibody System Dynamics* **12**(2), 147–172.

Maplesoft (2016) [2016a] URL `www.maplesoft.com/products/maplesim/`.

Maplesoft (2016) [2016b] URL `www.maplesoft.com/products/maple/`.

Massaro, M., Lot, R. and Cossalter, V. [2011] On engine-to-slip modelling for motorcycle traction control design, *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* **225**(1), 15–27.

Mechanical Simulation (2016) [2016] URL `www.carsim.com/`.

Meirovitch, L. [1970] *Methods of Analytical Dynamics* (McGraw-Hill, New York).

Motion Genesis (2016) [2016] URL `www.motiongenesis.com`.

MSC (2016) [2016] URL `www.mscsoftware.com/it/product/adams`.

Pacejka, H. B. [2006] *Tyre and Vehicle Dynamics*, 2nd ed. (Butterworth-Heinemann, Oxford), ISBN UK ISBN-13: 980-0-7506-6918-4.

PTC (2016) [2016] URL `www.sdfast.com/`.

Rosenthal, D. [1988] Triangulization of equations of motion for robotic systems, *J. Guid. Control Dyn.* **11**(3).

Samin, J. and Fisette, P. [2003] *Symbolic modeling of multibody systems* (Springer).

Sayers, M. [1990] *Symbolic computer methods to automatically formulate vehicle simulation codes*, PhD thesis, The University of Michigan.

Sayers, M. [1991] Symbolic computer language for multibody systems, *J. Guid. Control Dyn.* **14**(6), 11531163.

Sayers, M. [1993] Autosim, *Vehicle System Dynamics* **22**, 53–56.

Schiehlen (ed.) [1990] *Multibody Systems Handbook* (Springer).

Shi, P. and McPhee, J. [2002] Symbolic programming of a graph-theoretic approach to flexible multibody dynamics, *Mech. Struct. Mach.* **30**(1).

Siemens (2016) [2016] URL `www.plm.automation.siemens.com/en_us/products/lms/virtual-lab/motion`.

Simulia (2016) [2016] URL `www.simpack.com`.

Stuttgart University (2016) [2016] URL `www.itm.uni-stuttgart.de/research/neweul`.

Suh, H. and Radcliffe, C. [1978] *Kinematics and Mechanism Design* (Wiley, New York).

Tavernini, D., Massaro, M., Velenis, E., Katzourakis, D. and Lot, R. [2013] Minimum time cornering: The effect of road surface and car transmission layout, *Vehicle System Dynamics* **51**(10), 1533–1547, doi:10.1080/00423114.2013.813557.

Test set for IVP solvers [2017] URL `https://archimede.dm.uniba.it/~testset/testsetivpsolvers/?page_id=30`.

Torok, J. [2000] *Analytical Mechanics* (Wiley).

Wang, S.-L. S. [2001] Motion simulation with working model 2d and msc. visualnastran 4d, *Journal of Computing and Information Science in Engineering* .