Symbolic Dynamic Modelling of Locomotion Systems with Persistent Contacts - Application to the 3D Bicycle

Johan Mauny, Mathieu Porez, Frédéric Boyer

Symbolic Dynamic Modelling of Locomotion Systems with Persistent Contacts - Application to the 3D Bicycle.

Johan Mauny * **Mathieu Porez** * **Frédéric Boyer** *

* *LS2N - Automation, Production and Computer Sciences Department - IMT Atlantique Bretagne-Pays de la Loire - La Chantrerie, 4, rue Alfred Kastler, CS 20722, 44307 Nantes, Cedex 3, France. (e-mail: johan.mauny@mines-nantes.fr)*

Abstract: In this article, we propose a general symbolic dynamic modelling framework devoted to Mobile Multibody Systems subject to hard persistent contacts. In particular, all rigid planar and spatial wheeled vehicles belong to this class of systems. To illustrate the approach we apply it to a realistic model of the three dimensional bicycle. Though being a familiar object for everybody, deriving the fully nonlinear dynamics of this system in a closed symbolic form is far from being trivial. Using a Newton-Euler algorithm coupled to a projective approach based on an explicit model of the contacts, the approach is successfully applied to the simulation of a free bicycle. It shows how the passive asymptotic stabilisation of the bicycle can be naturally ensured when it is thrown with sufficient initial velocities.

Keywords: Dynamic modelling, nonlinear systems, nonholonomic systems, bicycle locomotion, dynamic stability.

1. INTRODUCTION

Since its birth at the beginning of the nineteen century, the bicycle has continuously evolved, from the first wooden draisine to our today's electric bicycle (Herlihy, 2004). Nowadays, it is back on the front stage as an icon of the ecological lifestyle and could be used to equip our future towns of fleets of partially-autonomous connected vehicles. Despite the fact that the bicycle is an everyday object which speaks to everyone, its modelling and fast simulation in the realistic three-dimensional context is far from being a child's play, especially when expecting a closed symbolic form of its fully nonlinear dynamics. This challenging issue is manifested by the long chain of contributions on the topic, starting from the earlier works by Bourlet (1899); Boussinesq (1899) to the recent results of Meijaard et al. (2007); Consolini and Maggiore (2013), including the important steps by Jones (1942); Henaff (1987); Franke et al. (1990) to name but a few. The reasons of these difficulties are numerous. First, in contrast to other familiar mechanical objects, as our industrial manipulators are, the bicycle is a non-holonomic system, i.e., a system whose description requires more parameters than the number of its local Degrees of Freedom (DoF). Mathematically, this translates into the fact that its dynamics are partially governed by a set of non-integrable constraints on the velocities. While these systems are known from a long time since the works of Hertz (Hertz, 1894), their numerical resolutions often requires to cope with a set of Differential Algebraic Equations (DAE) requiring specific methods of integration able to cope with stiff systems (Benner et al., 2015). In robotics, non-holonomic systems have been largely studied in the context of mobile robotics of planar wheeled vehicles such as the usual unicycles or the simplified model of car-like platforms (Campion et al., 1996). Though sharing non-holonomy with these simple wheeled platforms, the bicycle differs from them by the fact that contrarily to a simple unicycle, its locomotion dynamics cannot be fully described with a kinematic model but also require a further dynamic model. As such, the bicycle belongs to the less common class of dynamic non-holonomic systems which have been studied over the past years in the community of geometric mechanics and control (Bloch et al., 1996), with applications to planar systems as the snake-board (Ostrowski et al., 1995). Remarkably, in these systems, the locomotion is based on the transfer of kinetic momentums from their internal (shape) degrees of freedom to their external (net) ones, through non-sliding conditions imposed by their wheels (Ostrowski et al., 1995). However, the bicycle differs from these other dynamic non-holonomic locomotion systems, since its wheels are not used for kinetic momentum transfers, but rather to ensure the three-dimensional stability while rolling. Moreover, being essentially three-dimensional, its dynamics involve much more geometric nonlinearities than planar systems, making the modelling quite heavy and laborious. Finally, and above all these reasons, the geometric model of the contacts, i.e., the calculation of the contact point of the two wheels with the ground is not trivial at all, and in fact requires solving at each step of time, a nonlinear algebraic equation, a quartic, that prevents its dynamics to be stated in a closed symbolic form (Meijaard et al., 2007).

In the following, we address the issue of modelling the three-dimensional bicycle. To that end, we will immerse this objective in the more general context of modelling three-dimensional Mobile Multibody Systems (MMS) subject to persistent contacts with a planar solid ground.

Following (Boyer and Belkhiri, 2014), the geometric nonlinearities are reduced to their minimum by firstly deriving the dynamics of the MMS in their intrinsic form, i.e. directly on the configuration principal fiber bundle $SE(3) \times S$, where $SE(3)$ stands for the net displacements of the MMS, while S is the shape space of its internal DoF. Secondly, these free dynamics are projected onto the kernel of a set of constraints on $G \times S$ which model the rolling without slipping, non-sliding and non penetrating conditions imposed by the contacts between the wheels and the ground. To derive these equations (free dynamics and constraints) in a symbolic form, we will take advantage of the recursivity of Newton-Euler based algorithms (Featherstone, 2008), while calculating the kernel of the constraint equations with some of the symbolic functions of Matlab. The approach results in a single algorithm (given in appendix A) which can work for any tree-like MMS supported on the ground through persistent contacts. The algorithm is next applied to the case of a three-dimensional bicycle with comparisons with a benchmark recently proposed in (Meijaard et al., 2007). While addressing this special case, our general algorithm has to be fed with a model of the contact. Remarkably, the resolution of the quartic mentioned above is circumvented by moving the plane of the ground at each step of the simulation in order to ensure its point-contacts with the two wheels while keeping the gravity normal to the ground. This approach gives undetectable discrepancies with the numerical resolution of (Meijaard et al., 2007). These encouraging results tend to show that a closed form of the fully nonlinear dynamics of the three-dimensional bicycle could be reach in a near future.

The article is structured as follows. We start by giving the parametrization of a MMS and the notations we use in the article in section 2. In section 3, we quickly remind the Newton-Euler model of a MMS. In section 4, we introduce a Newton-Euler based algorithm solving the inverse dynamics of a MMS as they are required in the next section 5, which deals with the reduced dynamics of a MMS subject to persistent contacts. The general algorithm allowing to compute these reduced dynamics is given in section 6. This algorithm is applied to the three dimensional bicycle in section 7. The article ends by a conclusion (see section 8).

2. PARAMETRISATION AND NOTATIONS.

In all the following, we consider a Mobile Multibody System or MMS, i.e. a Multibody System subject to both overall rigid net motions in $SE(3)$ and shape time-variations in a manifold named shape-space. As represented in figure 1, the MMS has a tree-like structure interacting with its surrounding medium through hard persistent contacts as those imposed by rolling without slipping of a wheel, or the punctual contact of two solids. We attach to the ambient geometric space a fixed spatial orthonormed frame denoted by $\mathcal{F}_e = (O_e, s_e, n_e, a_e)$, where s_e supports the vertical axis, while (O_e, n_e, a_e) defines the ground which is assumed to be planar. The considered MMS is composed of a sequence of $n+1$ rigid bodies interconnected through n one Degree of Freedom (DoF) angular joints. Regarding the bodies, they are denoted $\mathcal{B}_0, \mathcal{B}_1, ..., \mathcal{B}_n$, where $\mathcal{B}_0$ stands for the reference body, i.e. an arbitrarily distinguished body

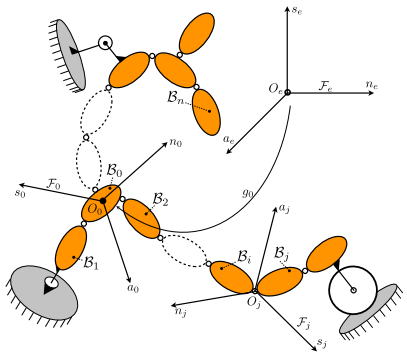

Fig. 1. Schematic view of a Mobile Multibody System.

whose motions define the net motions of the MMS, and with respect to which, the motions of the other bodies define the shape motions of the MMS. The bodies are numbered from $\mathcal{B}_0$ toward the tips of the branches in increasing order. In the following, we denote by j and i, the indices of the current body and its antecedent respectively, i.e., $i = a(j)$ where a denote the antecedent of j. We attach to each body $\mathcal{B}_j$ a mobile frame $\mathcal{F}_j = (O_j, s_j, n_j, a_j)$, where the center O_j coincides with the center of the joint j, and a_j supports the joint axis while s_j and n_j are defined according to the direct orientation of space. At any time t, the robot configuration is defined by the vector of joint positions $r = (r_1, ..., r_n)^T$ defining the relative angles around the joint axis between the bodies, together with the orientation matrix ${}^eR_0$ and the position vector ${}^eP_0$ of the mobile frame attached to the reference body $\mathcal{F}_0 = (O_0, s_0, n_0, a_0)$ with respect to $\mathcal{F}_e$. The time evolution of r defines the shape motion of the MMS, while that of $({}^eR_0, {}^eP_0) \triangleq {}^eg_0 \in SE(3)$ defines its rigid net motion. Regarding the contacts, they are modelled as a set of holonomic and/or non-holonomic kinematic constraints. Practically, these constraints are deduced by forcing the velocity of points of bodies in contact (e.g. with the ground) to be zero. If we define by m the number of independent constraints imposed by the contacts, in this paper, we will consider only the systems that fulfill the mobility condition $m < 6 + n$. Finally, throughout this article, we will use the following notational conventions. For any physical variable modelled by a tensor, the right lower index will represent the body index (to which it is related) while the left upper exponent will indicate the index of the reference frame, which is also the projection frame (e.g. ${}^eR_0$, ${}^eP_0$). When the tensor related to a body is expressed in the mobile frame of this body, the upper index is omitted. Finally, time derivation is sometimes denoted by a 'dot'.

3. NEWTON-EULER MODEL OF A MOBILE MULTIBODY SYSTEM.

To model our MMS, we use the Newton-Euler formalism of robotics, which works indifferently for MMS and MS (not mobile). Let us start by introducing the geometric model of the MMS which relates the pose of any frame $\mathcal{F}_j$ with that of the antecedent frame $\mathcal{F}_i$, both expressed in the earth frame $\mathcal{F}_e$ and represented by the two (4×4) matrices ${}^eg_i$ and ${}^eg_j$ of $SE(3)$. This model can be detailed as:

$$ {}^e g_j = {}^e g_i \, {}^i g_j(r_j) = {}^e g_i \begin{pmatrix} {}^i R_j(r_j) & {}^i P_j \\ 0 & 1 \end{pmatrix}, \quad (1) $$

where ${}^i R_j$ and ${}^i P_j$ are the orientation matrix and the position vector of $\mathcal{F}_j$ with respect to $\mathcal{F}_i$. Regarding the velocity of the body j, it is a (6×1) vector of $se(3)$ denoted η_j and related to the velocity of the antecedent body i through the recursive relation:

$$ \eta_j = (V_j^T, \Omega_j^T)^T = Ad_{{}^j g_i} \eta_i + \dot{r}_j A_j \ , \quad (2) $$

where V_j and Ω_j are respectively the linear and angular Galilean velocities of the considered body, both expressed in its mobile frame, $A_j = (0_3^T, a_j^T)^T$ is the (6×1) unit vector supporting the joint axis j, and $Ad_{{}^j g_i}$ is the adjoint map operator which allows changing a velocity in $se(3)$ from $\mathcal{F}_i$ to $\mathcal{F}_j$ (Murray et al., 1994):

$$ Ad_{{}^j g_i} = \begin{pmatrix} {}^j R_i & {}^j R_i {}^i \hat{P}_j^T \\ 0 & {}^j R_i \end{pmatrix}. \quad (3) $$

Let us remark that in (3), we introduced the 'hat' notation which changes a (3×1) vector into its associated (3×3) skew-symmetric tensor. Thus, for any vectors A and B in $\mathbb{R}^3$, $\hat{A}$ is defined such that $\hat{A}B = A \times B$.

Once the Galilean velocities are defined, by time derivation of (2), the acceleration of $\mathcal{B}_j$ is given by the relation:

$$ \dot{\eta}_j = Ad_{{}^j g_i} \dot{\eta}_i + \zeta_j + \ddot{r}_j A_j, \quad (4) $$

where ζ_j represents the component of accelerations in (4) which depends on velocities through the detailed expression:

$$ \zeta_j = \begin{pmatrix} ({}^j V_i + {}^j P_i \times {}^j \Omega_i) \times \dot{r}_j a_j \\ \dot{r}_j {}^j \Omega_i \times a_j \end{pmatrix}. \quad (5) $$

Finally, by applying the Newton's law and the Euler's theorem on the j^{th} body, one obtains the dynamic equations of $\mathcal{B}_j$ in the Newton-Euler form:

$$ f_j = \mathcal{M}_j \dot{\eta}_j + f_{in,j} + f_{ext,j} + \sum_{k|a(k)=j} Ad^T_{{}^k g_j} f_k, \quad (6) $$

where k are the indices of all the successive bodies to $\mathcal{B}_j$. Moreover, in (6), we introduced:

- the (6×1) force vector f_j (element of $se(3)^*$) exerted by $\mathcal{B}_i$ onto $\mathcal{B}_j$;
- the (6×6) inertia tensor $\mathcal{M}_j$ of $\mathcal{B}_j$ (element of $se(3)^* \otimes se(3)$), which can be detailed as:

$$ \mathcal{M}_j = \begin{pmatrix} M_j & MS_j^T \\ MS_j & I_j \end{pmatrix}, \quad (7) $$

  where M_j and I_j are the tensors of linear and angular inertia while MS_j is the tensor of first inertia moments, all being related to $\mathcal{B}_j$;
- the (6×1) vector of Coriolis and centrifugal forces:

$$ f_{in,j} = \begin{pmatrix} -\Omega_j \times (MS_j \Omega_j) + \Omega_j \times (M_j V_j) \\ \Omega_j \times (I_j \Omega_j) + MS_j (\Omega_j \times \dot{V}_j) \end{pmatrix}, \quad (8) $$

- the (6×1) vector of external forces denoted by $f_{ext,j}$ (like the gravity, the forces applied on the MMS by a fluid, etc ...).

4. A NEWTON-EULER ALGORITHM.

In the past years, the above Newton-Euler model has been extensively exploited to generate efficient and simple algorithms solving the inverse, forward and hybrid (inverse-forward) dynamics of both MS and MMS (Featherstone, 2008). These algorithms can be programmed either numerically or using customized symbolic programming techniques. Moreover, while they primarily enjoy a $o(n)$ complexity, they can be parallelized to obtain fast enhanced algorithms with a $o(log(n))$ complexity (Featherstone, 1999). When considering MMS, a simple Newton-Euler algorithm which will be used later, consists of computing the joint torques τ and the external reaction wrench f_0 exerted on the reference body $\mathcal{B}_0$, in order to impose a given motion $t \mapsto (r, \dot{r}, \ddot{r}, {}^e g_0, \eta_0, \dot{\eta}_0)(t)$. Such an inverse algorithm can be derived as follows. Firstly, we compute all the bodies transformations, velocities and accelerations $({}^0 g_j, \eta_j, \dot{\eta}_j)$ by using (1), (2) and (4) as forward recursions (from $\mathcal{B}_0$ to the tip branches) initialized by ${}^e g_0(t)$, $\eta_0(t)$ and $\dot{\eta}_0(t)$ and fed with $r(t)$, $\dot{r}(t)$, $\ddot{r}(t)$. Secondly, (6) is used as a backward recursion (from the tip branches to $\mathcal{B}_0$) for computing the inter-body wrench f_j. Finally, projecting f_j, $j = n, n-1, ..., 1$ onto the joint axis A_j gives τ, while the last computed wrench is merely f_0. This computational process is formally defined as:

$$ \begin{pmatrix} f_0 \\ \tau \end{pmatrix} = \mathcal{I}(\dot{\eta}_0, \eta_0, {}^e g_0, \ddot{r}, \dot{r}, r), $$

where $\mathcal{I}$ denotes the recursive inverse dynamics algorithm.

5. REDUCED LOCOMOTION DYNAMICS OF A MMS WITH PERSISTENT CONTACTS.

In accordance with the assumptions of section 2, we address the following dynamic problem: knowing at each time t, the state of the MMS $({}^e g_0, {}^e \eta_0, r, \dot{r})$, and the torques τ applied to the joints (through control laws or stress-strain material laws); the forward dynamic problem consists in calculating the accelerations of the reference body $\dot{\eta}_0$ of the MMS subject to hard persistent contacts, and the vector of joint accelerations $\ddot{r}$. In such a case, the forward dynamic equations can be written in the following assembled Lagrangian form (Boyer and Belkhiri, 2014):

$$ \begin{cases} \begin{pmatrix} \mathcal{M}_0^+ & M_0^{+T} \\ M_0^+ & m^+ \end{pmatrix} \begin{pmatrix} \dot{\eta}_0 \\ \ddot{r} \end{pmatrix} + \begin{pmatrix} F_0^+ \\ Q^+ - \tau \end{pmatrix} = \begin{pmatrix} A^T \\ B^T \end{pmatrix} \lambda, \\ {}^e \dot{g}_0 = {}^e g_0 \eta_0. \end{cases} \quad (9) $$

In (9), the bottom row stands for a reconstruction equation allowing to recover the motion of $\mathcal{B}_0$ on $SE(3)$ knowing the time evolution of its velocity η_0 in $se(3)$. The time evolution of η_0 along with that of the internal DoF of the MMS is governed by the top row of (9), in which $\mathcal{M}_0^+$ is the rigid inertia matrix of the MMS, M_0^+ is the coupling matrix between the external and internal DoFs and m^+ is the inertia matrix of the internal DoFs. As far as $(F_0^+, Q^+)^T$ and τ are concerned, they are the vector of the external and internal generalized forces and the vector of torques applied on the joints respectively. Finally, λ is the vector of Lagrange multipliers which physically represent the external reaction forces transmitted from the media to the MMS through the persistent contacts which are modelled by m independent kinematic constraints of the following general form:

$$ 0_m = A({}^e g_0, r)\eta_0 + B({}^e g_0, r)\dot{r}, \quad (10) $$

where, A and B are matrices ($m = rank(A, B)$) deduced from zero relative velocities conditions imposed at the contact points by the media to the MMS.

Using these constraints, the dynamics of the mobile system can be reduced through the projection of (9) in the kernel of the constraints (10). Practically, this reduction consists, firstly, to reduce the kinematics as follow:

$$\begin{pmatrix}\eta_0\\ \dot{r}\end{pmatrix} = H\begin{pmatrix}\eta_r\\ \dot{r}_r\end{pmatrix}, \text{ and} \tag{11}$$

$$\begin{pmatrix}\dot{\eta}_0\\ \ddot{r}\end{pmatrix} = H\begin{pmatrix}\dot{\eta}_r\\ \ddot{r}_r\end{pmatrix} + \dot{H}\begin{pmatrix}\eta_r\\ \dot{r}_r\end{pmatrix}. \tag{12}$$

where $H = \ker(A, B)$ is the kernel of the subspace of admissible velocities, $(\eta_r^T, \dot{r}_r^T)^T$ is the vector of reduced velocities. Secondly, introducing (12) in (9) that we then project onto the space of (virtual) velocities verifying (11), and since $H^T(A, B)^T = 0$, we obtain the reduced dynamics:

$$\begin{pmatrix}\dot{\eta}_r\\ \ddot{r}_r\end{pmatrix} = -\begin{pmatrix}\mathcal{M}_r^+ & M_r^{+T}\\ M_r^+ & {m_r}^+\end{pmatrix}^{-1}\begin{pmatrix}F_r^+\\ Q_r^+ - \tau\end{pmatrix}, \tag{13}$$

which have to be supplemented with the reduced reconstruction equation:

$$\begin{pmatrix}{}^e\dot{g}_0\\ \dot{r}\end{pmatrix} = \begin{pmatrix}{}^e g_0 & 0\\ 0 & 1\end{pmatrix} H\begin{pmatrix}\eta_r\\ \dot{r}_r\end{pmatrix}. \tag{14}$$

In (13), we have introduced the following reduced matrices:

$$\begin{pmatrix}\mathcal{M}_r^+ & M_r^{+T}\\ M_r^+ & {m_r}^+\end{pmatrix} = H^T\begin{pmatrix}\mathcal{M}_0^+ & M_0^{+T}\\ M_0^+ & m^+\end{pmatrix} H, \text{ and} \tag{15}$$

$$\begin{pmatrix}F_r^+\\ Q_r^+\end{pmatrix} = H^T\left(\begin{pmatrix}F_0^+\\ Q^+\end{pmatrix} + \begin{pmatrix}\mathcal{M}_0^+ & M_0^{+T}\\ M_0^+ & m^+\end{pmatrix}\dot{H}\begin{pmatrix}\eta_r\\ \dot{r}_r\end{pmatrix}\right). \tag{16}$$

6. COMPUTATIONAL ALGORITHM.

In this section, we propose a symbolic off-line computation of the reduced dynamics (13,14). It is based on the calculation of the free dynamics of the system, i.e. (9) with $\lambda = 0$, together with a recursive symbolic calculation of the constraints (10). Then, once all the matrices of (9) and (10) known, H and $\dot{H}$ and finally (13,14) are symbolically computed. Further details about these calculations are given in the three subsections below.

6.1 Calculation of the free dynamics

As regards the calculation of the free dynamics, it is based on the fact that one can reconstruct the assembled Lagrangian dynamics (9) by using the Newton-Euler inverse algorithm $\mathcal{I}$ of section 2.3. In fact, remarking that we have the identity:

$$\begin{pmatrix}\mathcal{M}_0^+ & M_0^{+T}\\ M_0^+ & m^+\end{pmatrix}\begin{pmatrix}\dot{\eta}_0\\ \ddot{r}\end{pmatrix} + \begin{pmatrix}F_0^+\\ Q^+\end{pmatrix} = \mathcal{I}(\dot{\eta}_0, \eta_0, {}^e g_0, \ddot{r}, \dot{r}, r),$$

it becomes straightforward to show that feeding $\mathcal{I}$ with some specific unitary inputs allows to calculate (numerically or symbolically), column after column, all the matrices $\mathcal{M}_0^+$, M_0^+, m^+ together with F_0^+ and Q^+.

6.2 Calculation of the constraints

To calculate the model of constraints (10), we start by defining for each body $\mathcal{B}_j$, its Jacobian matrix $\mathcal{J}_j$ as follows:

$$\eta_j = \mathcal{J}_j\begin{pmatrix}\eta_0\\ \dot{r}_j\end{pmatrix} \triangleq ({J_j}^{ext}, {J_j}^{int})\begin{pmatrix}\eta_0\\ \dot{r}_j\end{pmatrix}.$$

Referring to (2), these Jacobian matrices are calculated through the following forward recursion on the body indices:

$$\begin{cases}{J_j}^{ext} = Ad_{j_{g_i}}{J_j}^{ext},\\ {J_j}^{int} = Ad_{j_{g_i}}{J_i}^{int} + A_j e_j^T,\end{cases}$$

starting from the initial values ${J_0}^{ext} = 1_{6\times 6}$, ${J_0}^{int} = 0_{6\times(n-1)}$ and with e_j a $(n \times 1)$ unit vector whose the j^{th} component is equal to 1. Then, the algorithm properly calculates the matrices A and B of the model of constraints (10). This is performed by forcing to zero the velocity of the material point whose the position overlaps ${}^0P_{j(k)}$ at t in the direction ${}^0u(k)$ (which is an unit vector of $\mathcal{F}_0$). In these conditions, ${}^0P_{j(k)}$ and ${}^0u(k)$, with k the index of the constraint ($k \in \{1, 2...m\}$), are some inputs of the algorithm that have to be specified by the geometry of contacts. Finally, we initialize the matrix of constraints by:

$$(A, B) = 0_{m\times(6+n-1)},$$

and calculate from $k = 1$ to m the velocity of the material point whose the position overlaps $P_{j(k)}$:

$$V_{j(k)} = \begin{pmatrix}1_3 & -\hat{P}_{j(k)}\\ 0_3 & 1_3\end{pmatrix}\mathcal{J}_{j(k)},$$

that we project onto ${}^{j(k)}u(k)$, to derive the k^{th} row of the matrix of kinematic constraints (A, B):

$$(A, B)_{(k)} = {}^{j(k)}u(k)^T V_{j(k)}.$$

In this way, the matrix of kinematic constraints is built row by row.

6.3 Calculation of the reduced dynamics

Finally, H is computed symbolically using the Matlab function `Null` based on the Singular Value Decomposition, while $\dot{H}$ is obtained through a symbolic time-differentiation of H using the `diff` function of Matlab. At the end, using the free dynamics together with these expressions of H and $\dot{H}$, the reduced dynamics (13) are computed by using the symbolic expressions of the projections (15) and (16). The entire algorithm including the calculation of the free dynamics is reported in Appendix through 3 recursions.

7. APPLICATION TO THE BICYCLE.

In this section, we propose to apply the framework, introduced previously, to the simulation of the asymptotic stability of a free bicycle and to a controlled turning maneuver (see video https://youtu.be/B9mu7xQlL2w).

7.1 The bicycle.

The bicycle consist of four bodies: a frame, a fork and two wheels ($n = 4$), described in accordance with the assumptions presented in section 2. We define by w and c the wheel base length and the trail respectively. The steer axis tilt, angle between the fork and the vertical, is denoted by θ. As depicted in figure 2, the frame has been chosen as the reference body and, consequently, is denoted by $\mathcal{B}_0$. Starting from $\mathcal{B}_0$, the fork and the front wheel will be respectively denoted $\mathcal{B}_1$ and $\mathcal{B}_2$ while the rear wheel will be denoted $\mathcal{B}_3$. For each bodies of the bicycle, the

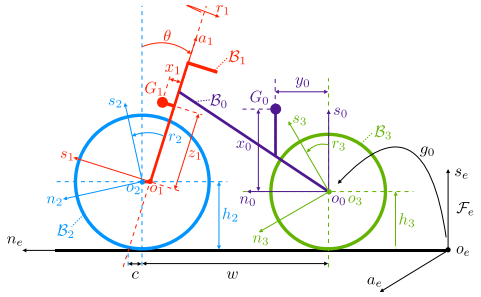

Fig. 2. Schematic view of a bicycle.

parameters are given with respect to the body's frame and the inertia matrices are express at the center of mass. The bicycle frame has a mass m_0 and its center of mass, G_0 is located in $\mathcal{F}_0$ at $(x_0, y_0, 0)^T$. Its inertia matrix expressed at G_0 in $\mathcal{F}_0$ is equal to:

$$\begin{pmatrix} I_{0xx} & I_{0xy} & 0 \\ I_{0xy} & I_{0yy} & 0 \\ 0 & 0 & I_{0zz} \end{pmatrix}.$$

As the fork is concerned, its center of mass is denoted G_1 which is located in $\mathcal{F}_1$ at $(x_1, 0, z_1)^T$. It has a mass m_1 and its inertia matrix is equal to:

$$\begin{pmatrix} I_{1xx} & 0 & I_{1zx} \\ 0 & I_{1yy} & 0 \\ I_{1zx} & 0 & I_{1zz} \end{pmatrix}.$$

In order to complet the parametrization of the studied system, the front and rear wheels have a mass denoted m_2 and m_3 and a radius denoted h_2 and h_3 respectively. Their center of mass is located at the joint axis (i.e at the wheel hubs) and due to their geometries, their inertia matrix is diagonal with, for the front wheel (for the rear wheel), an axial inertia moment I_{2zz} (I_{3zz}) and radial inertia moments $I_{2xx} = I_{2yy}$ ($I_{3xx} = I_{3yy}$ respectively).

7.2 Model of contacts.

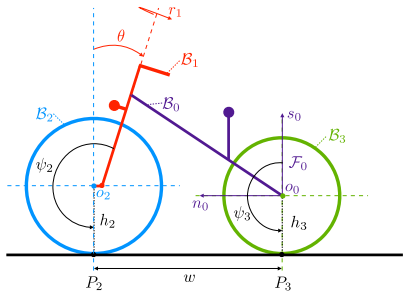

Fig. 3. Parametrization of contacts between the wheels and the ground.

We assume that the contacts between the wheels and the flat ground are perfect, i.e. the tires are not subjected to friction and deformation. In this example, the constraints reflect the fact that in any direction normal to the plane, each wheel can not penetrate nor separate the ground at the contact point while in a direction parallel to their plane, the wheels roll without slipping. In order to write the six non-holonomic rolling constraints under the form (10), we define two geometric contact points denoted by P_2 for the front wheel and P_3 for the rear wheel as illustrated in figure 3. To locate these points, we introduce two angles denoted by ψ_2 and ψ_3 which parameterize their angular positions along the boundaries (the tires) of the two wheels. The calculation of these contact points is not a trivial task since it requires to solve numerically a quartic at each time step of the simulation (Meijaard et al., 2007). Here this implicit calculus is replaced by an explicit calculation based on the projection of the dynamics in the kernel of the kinematic constraints including the two holonomic constraints imposed by no penetrating nor lifting. Thus, these two implicit constraints have to be solved once, before starting the simulation, in order to initialize it with a compatible configuration. In this context, for computing ψ_2 and ψ_3, we assume that the contacts are imposed on the material points of tires whose altitude (with respect to $\mathcal{F}_e$) is minimum, i.e.:

$$\frac{\partial({}^eP_2^T s_e)}{\partial \psi_2} = 0 \text{ and } \frac{\partial({}^eP_3^T s_e)}{\partial \psi_3} = 0,$$

where ${}^eP_2$ and ${}^eP_3$ are the positions of P_2 and P_3 in the earth frame $\mathcal{F}_e$ respectively, when s_e is the vertical axis. Once all calculations done, ψ_2 and ψ_3 are given by:

$$\psi_2 = \text{atan}(a_2/b_2) \text{ and } \psi_3 = \text{atan}(a_3/b_3), \qquad (17)$$

with:

$$\begin{cases} a_2 = \cos r_1 ({}^eR_0^{11} \sin\theta + {}^eR_0^{12} \cos\theta) + {}^eR_0^{13} \sin r_1, \\ b_2 = {}^eR_0^{11} \cos\theta - {}^eR_0^{12} \sin\theta, \\ a_3 = {}^eR_0^{12}, \text{ and} \\ b_3 = {}^eR_0^{11}, \end{cases}$$

where ${}^eR_0^{ij}$ is the i,j component of ${}^eR_0$. Once, ψ_1 and ψ_2 computed, (17) allows defining the current position of the contact points in which the velocity with respect to the ground are forced to zero. In short, this defines the points ${}^0P_{j(k)}$ and the directions $u(k)$ that feed the algorithm for the calculation of the A and B matrices of (10). Once H is calculated symbolically, the symbolic calculation of $\dot{H}$ needs to time-differentiate (17) using standard composition of derivations fed by the time differential of the arguments of (17):

$$\dot{\psi}_2 = \frac{\dot{a}_2 b_2 - \dot{b}_2 a_2}{a_2^2 + b_2^2}, \text{ and } \dot{\psi}_3 = \frac{\dot{a}_3 b_3 - \dot{b}_3 a_3}{a_3^2 + b_3^2},$$

with:

$$\begin{cases} \dot{a}_2 = ({}^eR_0^{12}\Omega_{0z} - {}^eR_0^{13}\Omega_{0y}) \cos r_1 \sin\theta \\ \quad +({}^eR_0^{13}\Omega_{0x} - {}^eR_0^{11}\Omega_{0z}) \cos r_1 \cos\theta \\ \quad +({}^eR_0^{11}\Omega_{0y} - {}^eR_0^{12}\Omega_{0x}) \sin r_1 \\ \quad -{}^eR_0^{11}\dot{r}_1 \sin r_1 \sin\theta - {}^eR_0^{12}\dot{r}_1 \sin r_1 \sin\theta \\ \quad +{}^eR_0^{13}\dot{r}_1 \cos r_1, \\ \dot{b}_2 = ({}^eR_0^{12}\Omega_{0z} - {}^eR_0^{13}\Omega_{0y}) \cos\theta \\ \quad -({}^eR_0^{13}\Omega_{0x} - {}^eR_0^{11}\Omega_{0z}) \sin\theta, \\ \dot{a}_3 = -{}^eR_0^{11}\Omega_{0z} + {}^eR_0^{13}\Omega_{0x}, \text{ and} \\ \dot{b}_3 = {}^eR_0^{12}\Omega_{0z} - {}^eR_0^{13}\Omega_{0y}. \end{cases}$$

where Ω_{0x}, Ω_{0y} and Ω_{0z} are the three components of the Galilean angular velocity of $\mathcal{B}_0$ in $\mathcal{F}_0$.

7.3 Results and Discussions

In order to simulate the bicycle, we applied the algorithm of Appendix A with the help of the Symbolic Toolbox of Matlab to generate an explicit analytical model of the bicycle free dynamics and its constraints. This model

Table 1. Simulation parameters.

Parameter	Value	Parameter	Value
m	6	n	4
w	1.02 m	g	9.81 m s^{-2}
c	0.08 m	$I_{2xx} = I_{2yy}$	0.1405 kg m^2
θ	$\pi/10$ rad	I_{2zz}	0.28 kg m^2
h_2	0.35 m	$I_{3xx} = I_{3yy}$	0.0603 kg m^2
h_3	0.3 m	I_{3zz}	0.12 kg m^2
x_0	0.6 m	I_{0xx}	2.8 kg m^2
y_0	0.3 m	I_{0yy}	9.2 kg m^2
x_1	0.0288 m	I_{0zz}	11.0 kg m^2
z_1	0.368 m	I_{0xy}	-2.4 kg m^2
m_0	85 kg	I_{1xx}	0.0584 kg m^2
m_1	4 kg	I_{1yy}	0.06 kg m^2
m_2	3 kg	I_{1zz}	0.0076 kg m^2
m_3	2 kg	I_{1zx}	-0.0091 kg m^2

depends on the 26 parameters of the system (see table 1). Then, using $H = \ker(A, B)$ and $\dot{H}$, the projection formula (15) and (16) are applied to the free dynamics to automatically generate the reduced dynamics (13), (14). A crude implementation of the algorithm (with no optimizations) provides a symbolic dynamic model that involves few thousand basic operations $(=, -, +, \times, /)$. Using a quaternion parametrization for reconstruction, the model is time-integrated numerically in Matlab with a predictor-corrector method (a fourth-order explicit method for the prediction step and a fifth-order implicit method for the correction step). At the end, the bicycle can be simulated in real time (with Matlab), on a intel i7 CPU @ 3 GHz (as example 5 s of simulation takes less than 2.6 s of computation with a time step of 0.005 s).

Passive asymptotic stabilisation of a bicycle: For this first example, we have chosen to reproduce the simulation presented in (Meijaard et al., 2007). It is related to the asymptotic stabilisation of a free bicycle (i.e. $\tau_1 = \tau_2 = \tau_3 = 0$). To that end, the initial conditions are chosen as follows. The speed along n_0 (forward axis) is fixed to $V_{0y} = 4.6\,\mathrm{m\,s^{-1}}$ while the angular speed around n_0 (roll axis) is fixed to $\Omega_{0y} = 0.5\,\mathrm{rad\,s^{-1}}$. A top view of the bicycle motion is displayed in figure 4. Figure 5 presents

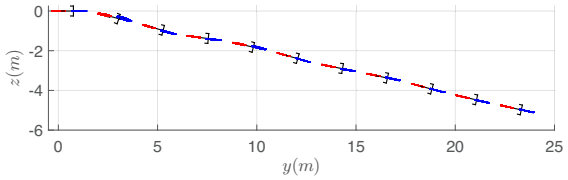

Fig. 4. Trajectory of the bicycle in the y-z plane, in the context of the asymptotic stabilisation study case. The time interval between each snapshoot is equal to 0.5 s

the plots of the time evolution of the rolling velocity $\Omega_0 y$ and forward velocity V_{0x} of the frame as well as the angular velocity of the handlebar $\dot{r}_1$ for the same test. Starting from their initial values, the plots of figure 5, which perfectly fit with the benchmark of (Meijaard et al., 2007), clearly show that the bicycle self-stabilizes along time. Moreover, it can be noticed that, while the time runs, the values of Ω_{0y} and $\dot{r}_1$ tend to zero while the forward speed of the frame increases up to reach a constant value closed to 4.625 m s^{-1} which is greater than the initial condition. As mentioned by (Meijaard et al., 2007), this is due to the energetic conservation of the system. Figure 6 and 7

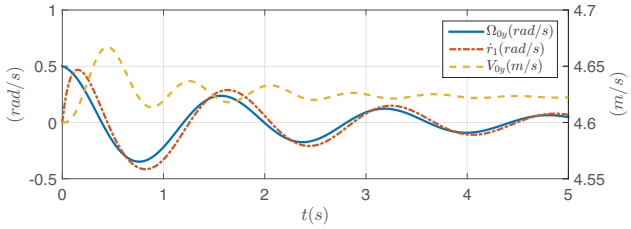

Fig. 5. The rolling speed Ω_{0y}, the forward speed V_{0y} and the handlebar speed $\dot{r}_1$ versus time t.

show the angular positions of contacts ψ_2 and ψ_3 on the front and rear wheels respectively. In the standing position of the bicycle (that plotted in figure 3), the contact points are located just behind the wheel axis (i.e. $\psi_2 = (\pi + \theta)$rad and $\psi_3 = \pi$ rad). Along simulation, these contact points first move before converging to their standing position (indicated by the red dotted line in the figures) when the bicycle recover its stable vertical configuration.

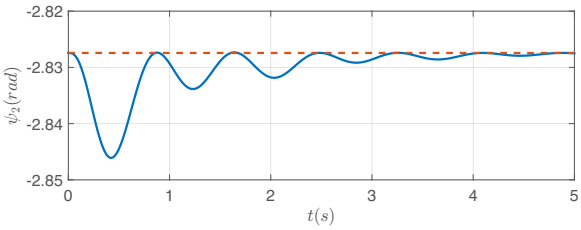

Fig. 6. Time-evolution of the angular position of the contact point on the front tyre (i.e. ψ_2) for the asymptotic stabilisation study case.

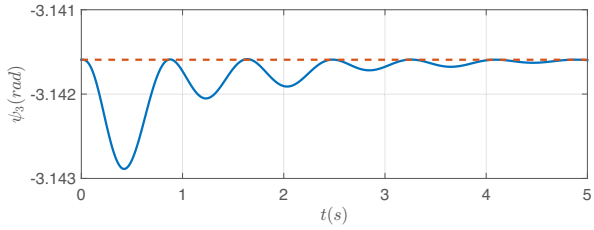

Fig. 7. Time-evolution of the angular position of the contact point on the rear tyre (i.e. ψ_3) for the asymptotic stabilisation study case.

The simulator offers also the possibility to compute the Lagrange multipliers associated to the constraints.In our case, there are six multipliers (3 per wheel). As an example, figures 8 shows the time evolution of λ_4 which represents the force applied radially (i.e. along the axis s_0) on the rear wheel at the contact point. This reaction force ensures the non-penetration of the rear tyre into the ground. Let us note that the multipliers are all expressed in the frame $\mathcal{F}_0$ (the bicycle frame) but can be easily expressed into the ground frame for the purpose of ground-tyres interactions study.

Controlled turn manoeuver: In this second and last numerical example, the handlebar and the rear wheel are actuated in order that the bicycle performs a turn at a

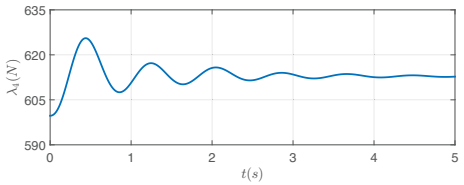

Fig. 8. The time evolution of λ_4 the force applied on the rear wheel prohibiting the no penetration of the tyre in the ground.

controlled forward speed. To do that, we use the following simple control torque law:

If $t \leq t_3$:

$$\tau_1(t) = -K_{ps}(r_{1d} f_s(t, t_1, t_2) - r_1(t)). \tag{18}$$

Else:

$$\tau_1(t) = -K_{ps}(r_{1d}(1 - f_s(t, t_3, t_4)) - r_1(t)) \ . \tag{19}$$

End.

where, K_{ps} is a steering proportional gain, r_{1d} is the desired steering angle, $t_1 \leq t_2 \leq t_3 \leq t_4$ are switching times used to perform a curved path, and $f_s(t, t_i, t_f)$ is a slope function defined as follows:

If $t \leq t_i$:

$$f_s = 0 \ . \tag{20}$$

Else if $t_i < t \leq t_f$:

$$f_s = \frac{t - t_i}{t_f - t_i} - \frac{1}{2\pi} \sin\left(2\pi \frac{t - t_i}{t_f - t_i}\right) \ . \tag{21}$$

Else:

$$f_s = 1 \ . \tag{22}$$

End.

In the above control law, t_i and t_f are the starting and ending times of the slope function respectively. As far as the rear wheel is concerned, the torque applied on its hub is defined by:

$$\tau_3 = K_{pd}(\dot{r}_{3d} - \dot{r}_3) \ , \tag{23}$$

where K_{pd} stands for a drive proportional gain and $\dot{r}_{3d}$ is the desired rear wheel speed. The values of the control law parameters are indicated in table 2. As shown in figure

Table 2. The simulation parameters of the controlled turn manoeuver.

Parameter	Value	Parameter	Value
K_{ps}	10 Nm	r_{1d}	$\pi/12$ rad
K_{pd}	40 Nms	$\dot{r}_{3d}$	20 rad.s^{-1}
t_1	0 s	$t_2 = t_3$	1 s
t_4	2 s		

9, the bicycle, starts from a straight vertical configuration with an initial forward velocity $V_{0y} = \dot{r}_{3d} h_3 = 6$ m.s^{-1}, and performs a turn of 0.067 m^{-1} curvature approximately. When the handlebar starts to move, the bicycle naturally (passively) tilts toward the center of the curve while turning. After a certain duration, the handlebar is straightened up. Moreover, the forward speed is enough high to ensure the asymptotic stabilisation of the bicycle (see Meijaard et al. (2007)). To illustrate this observation, in figure 10, we plotted for the same manoeuver the time evolution of the rolling, steering, and forward velocities which do stabilize with time. Figures 11 and 12 show the time evolution of the torques applied to the handlebar, i.e.

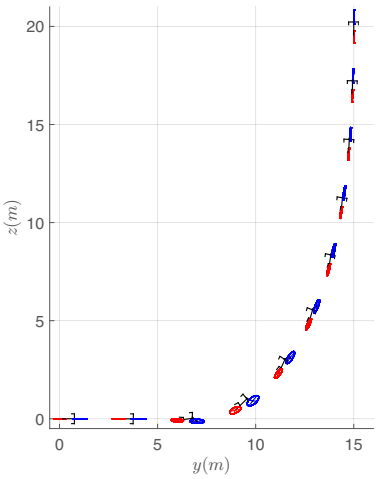

Fig. 9. Trajectory of the bicycle in the y-z plane when performing a controlled turn. The time interval between each snapshot is equal to 0.5 s.

τ_1, and to the rear hub, i.e. τ_3. Finally, as far as the contact points are concerned, we plotted in figure 13, the angular position of that between the front tire and the ground as a function of time. Note that during the maneuver, this contact point significantly migrates forward along the tire to reach 0.13 rad $\cong$ 7.5 deg.

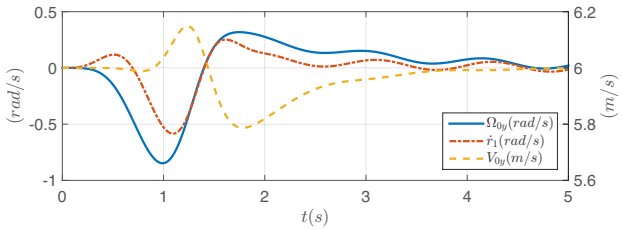

Fig. 10. Time-evolution of the rolling (in blue), steering (in red), and forward (in yellow) velocities during the turn manoeuver.

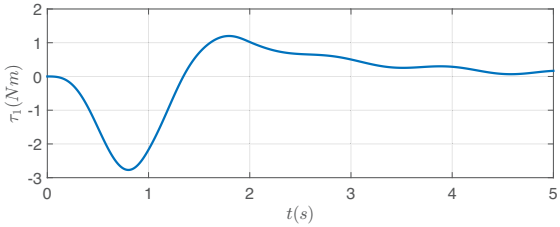

Fig. 11. Time-evolution of the steering torque τ_2.

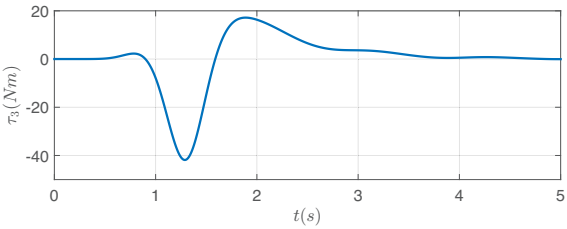

Fig. 12. Time-evolution of the driving torque τ_3.

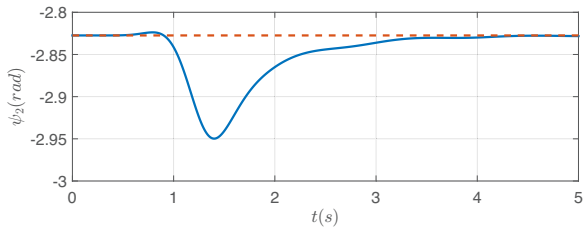

Fig. 13. Time-evolution of the angular position of the contact point on the front tyre (i.e. ψ_2) during the turn manoeuver.

8. CONCLUSIONS

In this paper, we have presented a general algorithm devoted to the symbolic modelling of the dynamics of tree-like structure Mobile Multibody System constrained by both holonomic and nonholonomic contacts. Based on the Newton-Euler approach of robot dynamics as well as the reduction process of (Boyer and Belkhiri, 2014), the proposed approach is applied to the three-dimensional bicycle with an explicit model of the contact. Futhermore, following Boyer and Belkhiri (2014), this dynamic model of the bicycle can be enhanced in order to take in account flexibilities. Comparisons with a recently proposed benchmark of this system give undetectable discrepancies and tend to prove that, contrarily to what is today believed, an exact and explicit model of the fully nonlinear dynamics of the bicycle in a closed form suited to nonlinear control could be derived in future.

REFERENCES

Benner, P., Bollhöfer, M., Kressner, D., Mehl, C., and Stykel, T. (2015). *Numerical algebra, matrix theory, differential-algebraic equations and control theory.* Springer International Publishing Switzerland.

Bloch, A.M., Krishnaprasad, P.S., Marsden, J.E., and Murray, R.M. (1996). Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis*, 136(1), 21–99.

Bourlet, C. (1899). Étude théorique sur la bicyclette. *Bulletin de la Société Mathématique de France*, 27, 76–96.

Boussinesq, J. (1899). Aperçu sur la théorie de la bicyclette. *Journal de Mathématiques Pures et Appliquées*, 117–136.

Boyer, F. and Belkhiri, A. (2014). Reduced locomotion dynamics with passive internal DoFs: Application to nonholonomic and soft robotics. *IEEE Transactions on Robotics*, 30(3), 578–592.

Campion, G., Bastin, G., and D'Andréa-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, 12(1), 47–62.

Consolini, L. and Maggiore, M. (2013). Control of a bicycle using virtual holonomic constraints. *Automatica*, 49(9), 2831–2839.

Featherstone, R. (1999). A divide-and-conquer articulated-body algorithm for parallel o(log(n)) calculation of rigid-body dynamics. part 2: Trees, loops, and accuracy. *The International Journal of Robotics Research*, 18(9), 876–892.

Featherstone, R. (2008). *Rigid Body Dynamics Algorithms.* Springer.

Franke, G., Suhr, W., and Riess, F. (1990). An advanced model of bicycle dynamics. *European Journal of Physics*, 11(2), 116–121.

Henaff, Y.L. (1987). Dynamical stability of the bicycle. *European Journal of Physics*, 8(3), 207–210.

Herlihy, D.V. (2004). *Bicycle: the history.* Yale University Press.

Hertz, H. (1894). *Die Prinzipen der Mechanik in neuem Zusammenhange dargestellt.* Gesamelte Werke, Band III. Leipzig.

Jones, A.T. (1942). Physics and bicycles. *Am. J. Phys.*, 10(6), 332.

Meijaard, J.P., Papadopoulos, J.M., Ruina, A., and Schwab, A.L. (2007). Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2084), 1955–1982.

Murray, R.M., Li, Z., and Sastry, S.S. (1994). *A mathematical introduction to robotic manipulation.* CRC press.

Ostrowski, J., Burdick, J., Lewis, A., and Murray, R.M. (1995). The mechanics of undulatory locomotion: The mixed kinematic and dynamic case. In *In Proc. IEEE International Conference on Robotics and Automation*, 1945–1951. Nagoya, Aichi, Japan.

Appendix A. ALGORITHM

$\mathcal{J}_0 = (1_{6\times6}, 0_{6\times(n-1)})$;
${J_0}^{int} = 0_{6\times(n-1)}$;
$Z_0 = 0_{6\times1}$;
for $j = 1$ **to** n **do**
  $\eta_j = Ad_{{}^jg_i}\eta_i + \dot{r}_j A_j$;
  $\mathcal{J}_j = Ad_{{}^jg_i}\mathcal{J}_i + A_j(0_6^T, e_j^T)^T$;
  ${J_j}^{int} = Ad_{{}^jg_i}{J_i}^{int} + A_j e_j^T$;
  $Z_j = Ad_{{}^jg_i}(\zeta_i + Z_i)$;
  $\mathcal{M}_j^+ = \mathcal{M}_j$;
  $F_j^+ = f_{in,j} + f_{ext,j}$;
  $M_j^+ = 0_{(n-1)\times6}$;
end
$\hat{m} = 0_{(n-1)\times(n-1)}$;
$Q^+ = 0_{(n-1)\times1}$;
for $j = n$ **to** 1 **do**
  $\mathcal{M}_i^+ = \mathcal{M}_i^+ + Ad^T_{{}^jg_i}\mathcal{M}_j^+ Ad_{{}^jg_i}$;
  $F_i^+ = F_i^+ + Ad^T_{{}^jg_i}(F_j^+ + \mathcal{M}_j^+\zeta_j)$;
  $M_i^+ = M_i^+ + Ad^T_{{}^jg_i}(M_j^+ + \mathcal{M}_j^+(A_j e_j^T))$;
  $\hat{m} = \hat{m} + e_j(A_j^T(\mathcal{M}_j^+ {J_j}^{int}))$;
  $Q^+ = Q^+ + e_j(A_j^T(F_j^+ + \mathcal{M}_j^+\zeta_j + \mathcal{M}_j^+ Z_j))$;
end
$m^+ = \hat{m} + \hat{m}^T - diag(\hat{m})$; $(A, B) = 0_{m\times(6+n-1)}$;
for $k = 1$ **to** m **do**
  $V_{j(k)} = \begin{pmatrix} 1_3 & -\hat{P}_{j(k)} \\ 0_3 & 1_3 \end{pmatrix} \mathcal{J}_{j(k)}$;
  $(A, B) = (A, B) + e_k\left(u_{j(k)}^T V_{j(k)}\right)$;
end