Symbolic derivation of control models for nonholonomic mechanical systems.

Paweł Cesar Sanjuan Szklarz
Institute of Aeronautics and Applied Mechanics
Warsaw University of Technology
00-665 Warsaw, Nowowiejska 24 St.
Email: pszklarz@meil.pw.edu.pl
Elżbieta Jarzębowska
Institute of Aeronautics and Applied Mechanics
Warsaw University of Technology
00-665 Warsaw, Nowowiejska 24 St.
Email: elajarz@meil.pw.edu.pl

Abstract

A systematic approach for a definition and derivation of control models for mechanical systems with nonholonomic constraints is presented. A mechanical structure of a system is translated to the geometric definition of local coordinate transformations. Further, mathematical equations of a control model can be matched to the geometric definition. Finally, numerical algorithms for numerical simulations are built as translations of an underlying structure to code procedures.

I. INTRODUCTION

At the conceptual level, the development of mechanical models is related to simple chains of translation and rotation transformations. In the presence of nonholonomic constraints, generating of constraint equations is realized by a distribution in the tangent bundle of the state manifold. In practice, this simplicity is not visible after the selection of specific coordinates and the translation of such models directly to ordinary differential equations. Such equations do not preserve the underlying mechanical system properties. A good example of this situation is the list of different bike models reported in the literature and the number of mistakes also reported for them; see for example [3–9, 12–14].

In this paper, a development of control models is based upon the Euler-Lagrange equations with reaction forces for nonholonomic constraints. To make use of the geometric structure of the mechanical model, we refer to the Levi-Civita connection of the metric related of the kinetic energy. For the constraints, a projective affine connection is presented; see [2, 10] for more details about this approach.

The novelty of the presented approach, is the symbolic generation of all model equations with the preservation of its underlying geometric structure. Model equations are presented as a composition of simple elements related to the mechanical model design. The structure of such composition is directly related to the geometry of the model. Also, the resulted equations are translated to numerical procedures ready for numerical simulations. The final numerical equations also preserve the geometric structure of the model.

The presented approach is then a systematic method for the development of control models for mechanical systems. Mistakes are possible only in a definition of the geometry of a model. All the subsequent elements and models are generated automatically in the computer. No special selection of bases for translations and rotations are necessary. The final equations are presented as a composition of simple elements that can be easily calculated and validated in any base. No commercial or close software is used. Software is base on Maxima, a Computer Algebra System and author's own procedures. The generation of a model of a bike is presented as an illustration of the theoretical development.

II. GEOMETRIC DEFINITION OF A MECHANICAL SYSTEM

For the definition of the mechanical models geometry, translation transformations T and rotations $B \in SO(3)$ are used. No special selection of bases for translations and rotations are necessary. The final equations are presented as a composition of simple elements that can be easily calculated and validated in any base.

A mechanical system can be represented as a directed graph of rigid bodies connected by design and material constraints. The development of such constraint equations is related to physical laws and system design. To obtain a system model equations,

URN: urn:nbn:gbv:ilm1-2014iwk:3

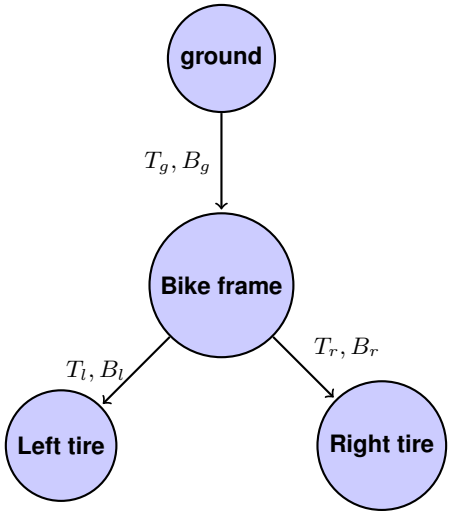


Fig. 1: Geometric definition of bicycle model elements.

it is necessary and sufficient to describe the geometry of the mechanical system design. Such geometric description is given by:

- **Connection graph** - An acyclic directed graph representing connections between rigid bodies that conform to the mechanical system.
- **Displacements** - On each arrow of the connection graph, a displacement transformation (rotation and translation) that expresses the relation between the bodies on the vertex.

Holonomic constraints can be often expressed as limitations to the space of available displacement and rotations. In such case, it is possible to select parameterization of the subspace of translations and rotations.

Note that displacements are assumed to be defined in the reference frame of the rigid body in the start of the arrow.

A. Definition of nonholonomic constraints

Nonholonomic constraints for mechanical systems are related to a no-slip condition of the instantaneous contact point between the ground and a mechanical system components. By a geometric definition of the mechanical system, the nonholonomic constraints are defined by a non-integrable distribution in the tangent bundle of the state space.

A general formula for the nonholonomic constraint equations is given by the classical equation

$$0 = v_o(q, \dot{q}) + \omega(q, \dot{q}) \times c(q), \tag{1}$$

where

- v_o is the linear velocity of a reference frame.
- Operator $\omega(q, \dot{q}) \times (\cdot) := \dot{B}(q, \dot{q}) B^{-1}(\cdot)$ is the angular velocity of a reference frame.
- $c(q) = B(q)C(q)$ is the position of the contact point with respect to the global reference frame orientation.

Because the position of the elements of the mechanical system model is the composition of several affine transformations, the general formula for nonholonomic constraints contains components that depend on all the elements in the connection chain. This general approach leads to very complex equations that cannot be easily decoupled into simpler components.

In the special case of elements with wheel geometry, special equations for the nonholonomic constraints can be found; see [11]. For the wheel geometry, the nonholonomic constraints can be expressed as

$$0 = v_o(q, \dot{q}) + r\dot{B}(q, \dot{q})(u_y \times (\frac{u_y \times B^{-1}(q) u_z}{\| u_y \times B^{-1}(q) u_z \|})). \tag{2}$$

where

- u_z, u_y are normal vectors in direction Z, Y in the local reference frame

III. Mechanical systems as affine connection control systems

Following [2], we develop mechanical system equations on the basis of an affine connection related to the kinetic energy and the distribution defined by the nonholonomic constraints.

The dynamics of the mechanical control system with the state space $\mathcal{Q}$ and constraints given by a $(n-l)$-dimensional distribution $\mathcal{D}$ on $\mathcal{Q}$. The Euler-Lagrange's equations of the system are given by:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \sum_{i=1}^{m} u_i(t) F^i + \sum_{j=1}^{l} \lambda^j \omega_j, \tag{3}$$

where, for a given Riemannian metric $\mathcal{G}$ and potential energy function V on $\mathcal{Q}$, $L(q,\dot{q}) = \frac{1}{2}\mathcal{G}(\dot{q},\dot{q}) - V(q)$ is the Lagrangian of the system.

Equation 3 can be written as

$$\begin{cases} \nabla^{\mathcal{G}}_{\dot{c}(t)}\dot{c}(t) = \lambda(t) - grad(V) + \sum_{i=1}^{m} u_i(t) Y^i(c(t)) \\ \dot{c}(t) \in \mathcal{D}_c(t), \end{cases} \tag{4}$$

where reaction forces λ^j are now seen as a section λ of $\mathcal{D}^{\perp}$ along the trajectory c. Let $\mathscr{P} : T\mathcal{Q} \to \mathcal{D}, \mathscr{Q} : T\mathcal{Q} \to \mathcal{D}^{\perp}$ denote the complementary $\mathcal{G}$ projectors, then the affine connection

$$\overline{\nabla}_X Y = \nabla^{\mathcal{G}}_X Y + (\nabla^{\mathcal{G}}_X \mathscr{Q})(Y) = \mathscr{P}(\nabla^{\mathcal{G}}_X Y) + \nabla^{\mathcal{G}}_X(\mathscr{Q}Y) \tag{5}$$

allow to rewrite equation 3 as

$$\overline{\nabla}^{\mathcal{G}}_{\dot{c}(t)}\dot{c}(t) = \mathscr{P}(grad(V)) + \sum_{i=1}^{m} u_i(t)\mathscr{P}(Y^i(c(t))) \tag{6}$$

The calculations of the $\mathcal{G}$ projectors is equivalent to the calculation of reaction forces of the nonholonomic constraints. In our generation algorithm, the related system of linear equations is generated and calculated online. This provides an implicit definition of final equations. Numerical experiments with different models show that such implicit form of the final equations is adequate for the development of control algorithms.

In this paper, we demonstrate how to compute all these transformations on the basis of the geometric definition of the mechanical system. As a result, set related procedures that reflect the structure of the geometric definition on the mechanical system are developed. This approach allows the generation of numerical codes for realization of numerical simulations.

IV. Energy functions calculation

The mechanical system is defined as a acyclic directed graph as expressed in the section II. To calculate the final model equations it is necessary to define the kinetic energy given as a Riemannian metric $\mathcal{G}$ in the state manifold and the potential energy V of the system.

Note that presented equations are well known, and the procedure is straightforward:

1) State manifold is given by the variables used to express the edges transformations.
2) The position of the system elements is calculated using forward kinematics.
3) Kinematical and potential energy are provided as a function of state variables.

To relevant information on the presented procedure is the form of the output equations. Symbolic equations are split to significant and repeating expressions that are calculated separately. Also numerical procedures calculations inherit the model structure to provide performance and accuracy.

A. Energy for a single node

To calculate the position of a selected node, forward kinematics can be used. Position of the local frame reference is given by the composition of transformations in the connection chain on the selected node. The information that express the node position is then given by the chain of transformations:

$$[T_1, B_1], [T_2, B_2], \ldots, [T_n, B_n] \tag{7}$$

Rotation of the local reference frame is given by

$$B = B_1 B_2 \dots B_n \tag{8}$$

Position of the local frame center is

$$c = T_1 + B_1(T_2 + B_2(\cdots + B_{n-1}(T_n)\dots) \tag{9}$$

Calculation of the node linear velocity is obtained by derivation of equation (9). This calculations can be evaluated very efficiently on any computer algebra system. The final result is complex for hand manipulation, but can be easily translated to numerical code. Simple optimization can be archived by separation of repeating trigonometric expressions to avoid expensive instructions during simulation and implementation of final algorithms. Velocity is given as

$$v = \frac{dc}{dt} \tag{10}$$

Angular velocity is given by the equation

$$\omega \times \cdot = \frac{dB}{dt} B^{-1} \tag{11}$$

Direct calculation of the angular velocity using equation (11) is not effective. The resulting matrix representation contains zeros in the main diagonal and angular velocity vector values are duplicated in the matrix. Direct translation to symbolic equations and numerical code is possible. Such direct approach lead to instable numerical procedures.

To obtain a better representation of the angular velocity we expand the equation (11) using equation (8).

Note that

$$\begin{aligned} \frac{dB}{dt} &= \frac{dB_1}{dt} B_2 \dots B_n + B_1 \frac{dB_2}{dt} \dots B_n + \\ &\quad \cdots + B_1 B_2 \dots \frac{dB_n}{dt} \end{aligned} \tag{12}$$

and

$$B^{-1} = B_n^{-1} \dots B_2^{-1} B_1^{-1} \tag{13}$$

Substitute (12) and (13) in (11) to obtain

$$\begin{aligned} \omega\times &= \frac{dB_1}{dt} B_1^{-1} + B_1 \frac{dB_2}{dt} B_2^{-1} B_1^{-1} + \\ &\quad \cdots + B_1 B_2 \dots \frac{dB_n}{dt} B_n^{-1} \dots B_2^{-1} B_1^{-1} \end{aligned} \tag{14}$$

Note that terms in equation (14) have the general form

$$R \frac{dB_i}{dt} B_i^{-1} R^{-1} \tag{15}$$

that represent a local angular velocity calculation and a change of base.

Symbolic calculation of (14) can be calculated efficiently. Inverse of rotation matrix match the transpose matrix. Base changes are nested, so duplication of calculations can be used to optimize equations generation. For numerical code generation, it is important to reflect the equations symmetry on the procedures calculation.

Finally, the node kinetic energy is given by the equation

$$\frac{1}{2} m(v \cdot v) + \frac{1}{2} \omega \cdot \{I\} \cdot \omega \tag{16}$$

It is important to choose local reference frame to match the principal axes of inertia. Then equation (16) don't contain products of the type $\omega_i \omega_j$ where $i \neq j$. This is a important optimalization, because expressions for angular velocity ω are the most complex terms in equation (16).

Potential energy can be assumet to be a function of the reference frame center location

$$V(q) = F(c) \tag{17}$$

this include the case of the gravity field.

1) Single node derivative calculation: For the Euler-Lagrange equations the following partial derivatives expression must be calculated:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} = 0$$

A direct approach could be the generation of the symbolic expression related to this operation, and the calculation of numerical code. This approach lost the structure and symmetry related to the derivative operator. A optimal numerical prodecure must contain the structure of this derivatives in the steps used to calculate the final equations values. To translate the structure of the equation to the numerical procedures, the following properties are used

Coordinate separation
: Calculation for every state variable q_k are separated.

Derivative operator is linear
: Every top level addition expression is calculated independently.

Projection to matrix form
: Final equations are grouped to create matrix functions.

The projection to matrix form is related to the view of Euler-Lagrange equations as equations of the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = 0 \tag{18}$$

where mass matrix $M(q)$ is given by the Riemannian metric of the kinematic energy. Matrix $C(q,\dot{q})$ is defined by terms of the form

$$c_{kj} = \sum_{i=1}^{n} c_{ijk}\dot{q}_i \tag{19}$$

where c_{ijk} are the Christoffel symbols

$$c_{ijk} = \frac{1}{2}\left(\frac{\partial g_{ij}}{\partial q_k} + \frac{\partial g_{ik}}{\partial q_j} - \frac{\partial g_{jk}}{\partial q_i}\right).$$

B. Composition of nodes energy

Lets compare energy components of two interconnected nodes $\alpha \to \beta$ in the geometric definition.

$$\begin{aligned} c_\alpha &= T_1 + B_1(T_2 + B_2(\cdots + B_{n-1}(T_n)\dots) \\ c_\beta &= T_1 + B_1(T_2 + B_2(\cdots + B_{n-1}(T_n + B_n(T_{n+1}))\dots) \end{aligned}$$

$$\omega_\beta \times = \omega_\alpha \times + B_1 B_2 \dots B_n \frac{dB_{n+1}}{dt} B_{n+1}^{-1} B_n \dots B_2^{-1} B_1^{-1}$$

This equations clearly show a nested pattern in the equations of interconnected nodes. To exploit this structure, it is necessary to arrange the symbolic calculations and numerical procedures to reuse common parts in the given expressions.

In the case of the angular velocity, following equation (15) note that the extra term can be split to

$$K B_n \frac{dB_{n+1}}{dt} B_{n+1}^{-1} B_n K^{-1} \tag{20}$$

where $K = B_1 B_2 \dots B_{n-1}$. During symbolic and numerical calculations it is enough to remember references to such nested matrix K globally for all the model.

For linear velocity, note that calculation of c_α and c_β can be performed in parallel. Define a function

$$L_n(v) = T_1 + B_1(T_2 + B_2(\cdots + B_{n-1}(v)\dots)) \tag{21}$$

then we have

$$\begin{aligned} c_\alpha &= L_n(T_n) \\ c_\beta &= L_n(T_n + B_n(T_{n+1})). \end{aligned}$$

This provide to a recursive definition of function L_n. To optimize the calculation of L_n, all input values related to futher nodes in the geometric definition graph can be calculated, and then the linear transformations can be applied in parallel.

Note that the two nested patterns have a different nature. For angular velocity, calculations are optimal in graph direction and optimalization are realized reusing previously calculated terms. For linear velocity, calculations are optimal in inverse graph direction and are based on grouping input values for similar functions.

1) Derivative calculation for composed nodes: The energy terms of separate nodes are added independently in the final energy function. This means that derivation can be realized independently and the final matrix form is just the union of the single nodes matrices.

The presented dependencies in the internal representation of the nodes equations must be used to optimize derivative operations. This is easily realized following the nested patterns presented.

V. Nonholonomic constrains equations

Nonholonomic constrains are assumed to be the result of wheel components with contact to a flat rigid ground. To calculate such nonholonomic constraints, equation (2) is applied. For a complete derivation of equation (2) see [11].

$$0 = v_o(q,\dot{q}) + r\dot{B}(q,\dot{q})(u_y \times (\frac{u_y \times B^{-1}(q)u_z}{\|u_y \times B^{-1}(q)u_z\|})). \tag{2}$$

Wheel center can be assumed to match the center of the local reference frame. Then velocity term $v_o(q,\dot{q})$ match the calculations of equations (10).

To calculate the term related to wheel rotation

$$r\dot{B}(q,\dot{q})(u_y \times (\frac{u_y \times B^{-1}(q)u_z}{\|u_y \times B^{-1}(q)u_z\|})) \tag{22}$$

note that rotation matrix B is given by (8) and that the operation $\times$ on a vector is a simple operation on the input vectors:

$$u_y \times (X,Y,Z)^T = (-Z,0,X)^T. \tag{23}$$

Calculation of term $\dot{B}(q,\dot{q})$ can be realized as show in equation (12).

A. Reaction forces equations

Calculation of reaction forces is related to the affine connection (5). To calculate the reaction forces during simulation and control algorithms implementation, it is necessary to derive a numerical procedure from equation (36) by derivation over time:

$$\frac{d}{dt}(D(q,\dot{q})\dot{q}) = D(q,\dot{q})\ddot{q} + \frac{d}{dt}(D(q,\dot{q}))\dot{q} = 0. \tag{24}$$

Matrix $D(q,\dot{q})$ represent the nonintegrable distribution of the nonholonomic constraints. The symbolic calculation of $D(q,\dot{q})$ from equation (36) is possible, but the resulting equations are complex and can not by optimized automatically for calculations. Also, derivation of matrix $D(q,\dot{q})$ elements independently contain many duplicated nested terms that can not be extracted by algebraic simplification. Such duplication lead to have bad performace on directly generated numerical procedures.

A geometric interpretation of the complex normalized term is provided in subsection V-B. Then a optimal derivative calculation procedure is presented in subsection V-C

B. Geometric interpretation of normalized constraint term

The mayor obstacle in the generation of a usefull symbolic representation of the nonholonomic constrains (2) is the normalization in used in the term

$$\frac{u_y \times B^{-1}(q)u_z}{\|u_y \times B^{-1}(q)u_z\|}. \tag{25}$$

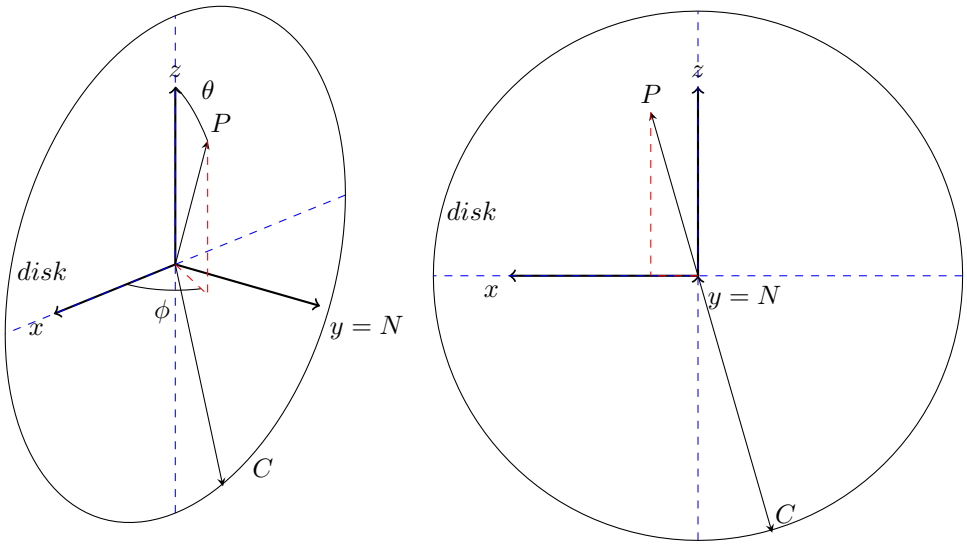
The geometric interpretation of term (25) is related to the position of the contact point with the ground in the wheel boundary. During wheel rotation, the contact point is located in the lowest point in ground normal vector direction. To find such point a projection of the ground normal vector to the wheel plane is realized and then a normalization is necessary to match wheel radius. Note that this normalization ratio is related to the relative tilting between the wheel plane and the ground plane. This is expressed in the basic relation for the cross product between vectors

$$a \times b = \|a\|\|b\|\sin(\tau)n$$

where n is the normal vector to the plane defined by a, b and τ is the angle between a, b.

Figure 2 shows this situation in the local reference frame of the wheel. Vector N is the normal vector to the wheel plane and vector P is the normal vector to the ground plane. C points to the contact point between wheel and the ground. The relavant angle τ in this setup is the angle between vector P and the wheel plane.

Vectors P and N are unitary, it is $\|P\| = 1$ and $\|N\| = 1$. Then, the calculation of $\sin(\tau)$ can be easily realized numerically.



(a) 3D projection of the wheel reference frame.

(b) 2D projection of the wheel reference frame to the plane xz.

Fig. 2: The ground normal vector P and the wheel normal vector N in wheel reference frame.

C. Optimal derivative calculation

The final goal is equation (24) is the calculation of a linear equation of the form

$$D(q,\dot{q})\ddot{q} = b(q,\dot{q}). \tag{26}$$

Linear equation (26) is necessary on the numerical level to calculate the reaction forces. In this subsection a numerical procedure for calculation of matrix $D(q,\dot{q})$ and vector $b(q,\dot{q})$ coeficients is developed.

Rewrite equation (2) as

$$0 = v_o(q,\dot{q}) + r\dot{B}(q,\dot{q})c(q) \tag{27}$$

where

$$c(q) := (u_y \times (\frac{u_y \times B^{-1}(q)u_z}{\|u_y \times B^{-1}(q)u_z\|})) \tag{28}$$

Derivation of (27) over time lead to

$$0 = \frac{dv_o(q,\dot{q})}{dt} + r\frac{d\dot{B}(q,\dot{q})}{dt}c(q) + r\dot{B}(q,\dot{q})\frac{dc(q)}{dt}. \tag{29}$$

Calculation of term

$$\frac{dv_o(q,\dot{q})}{dt} \tag{30}$$

can be handled at the symbolical level. Elements of the vector $v_o(q,\dot{q})$ are simple polynomial expressions of $\sin q_i, \cos q_i, \dot{q}_i$. After derivation, coeficient terms containing $\ddot{q}_i$ can be separated to split the result between matrix $D(q,\dot{q})$ and vector $b(q,\dot{q})$. Numerical calculation of resulting terms can be automatically optimized be a preprocessing of repeting trigonometrical terms.

Calculation of term

$$r\frac{d\dot{B}(q,\dot{q})}{dt}c(q) \tag{31}$$

is realized in two steps.

First, a numerical procedure to calculate vector $c(q)$ value is implemented. Note that terms in equations (28) can be easily calculated because $B^{-1} = B^T$, operation $B^{-1}u_z$ is a simple row selection and cross product $u_y \times \cdot$ is a simple linear operation as shown in (23).

Second, symbolic calculation of matrix

$$\frac{d\dot{B}(q,\dot{q})}{dt}$$

is realized. As previously, coeficient terms containing $\ddot{q}_i$ are extracted and a split between $D(q,\dot{q})$ and $b(q,\dot{q})$ is realized.

The final result for term (31) are two numerical procedures. First calculate $c(q)$ numerically, and the second calculate linear equation coeficients using the value of $c(q)$ as input.

Calculation of term

$$r\dot{B}(q,\dot{q})\frac{dc(q)}{dt} \tag{32}$$

requires a special procedure to avoid duplicated terms calculation. Note that term (32) have impact only in the value of vector $b(q,\dot{q})$. Then, direct values calculations can be performed and the result can be added to $b(q,\dot{q})$ independently of the previous terms.

Assume notation (x,y,z) for the vector $u_y \times B^{-1}(q)u_z$ in (28). Them the general form of the coordinates of the normalized vector is

$$\frac{x}{\sqrt{z^2+y^2+x^2}} \tag{33}$$

and derivation over time provide with the result

$$\frac{\dot{x}}{\sqrt{z^2+y^2+x^2}} - \frac{x\ (2\,z\,\dot{z}+2\,y\,\dot{y}+2\,x\,\dot{x})}{2\ (z^2+y^2+x^2)^{\frac{3}{2}}}. \tag{34}$$

Remember that matrix B represent the composition of all rotations matrices for the node in the geometric graph representation. Because wheel elements are modeled as leaf nodes in the geometric graph representation, then matrix B is the composition of all rotations from the root node to a leaf node. This means that coordinates (x,y,z) are very complex algebraic expression selected from matrix B^{-1} by the operation $u_y \times ((\cdot)u_z)$.

Independent calculation of each coordinate derivative as in (34), lead to repeting calculation of the terms

$$x, y, z, x^2+y^2+z^2, \frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}.$$

Algebraic or trigonometric simplifications don't simplify the final equations, because the transformation is related to the projection of the ground normal vector to the wheel plane.

Finally term (32) calculation is realized in the following steps. First, symbolic expressions for coordinates (x,y,z) are calculated. Second, symbolic derivatives

$$\frac{dx}{dt}, \frac{dy}{dt}, \frac{dz}{dt}$$

are calculated. Third, equations (34) are calculated numerically for coordinates x and z, y coordinate is ignored by $u_y \times \cdot$. Fourth, derivative value

$$\frac{dc(q)}{dt}$$

is calculated by application of operator $u_y \times \cdot$ to the previous result. Finally, the resulting vector is multiplied by the value of matrix $\dot{B}(q,\dot{q})$ and added to $b(q,\dot{q})$.

VI. Derivation of Euler-Lagrange's equations

On base of energy equations from section IV and nonholonomic constraint equations from section V the final Euler-Lagrange's equations will be calculated.

As defined in section III the Euler-Lagrange's equations with nonholonomic constraint are defined by a affine connection constructed from the Riemannian metric and the nonholonomic constraints distribution sections. The calculation of the complementary $\mathcal{G}$ projectors

$$\mathscr{P} : T\mathcal{Q} \to \mathcal{D}, \mathscr{Q} : T\mathcal{Q} \to \mathcal{D}^{\perp}$$

can be expressed as a set of linear equations for the reaction forces λ_i. For every wheel in the system, a set of 3 reaction equations is created. Analytic resolution of the reaction forces equations is not effective, because every single term in the equation is a complex expression with potential dependencies in every state variable. Numerical resolutions of the reaction

forces equations is fast and effective, because the equations dimention is small. The biggest numerical cost is related to the calculation of equations terms values.

It is expected that reaction forces will have a continuous behavior, so iterative methods for the solution of reaction equation forces can be utilized for performance. After a state variable update during simulation, the previous reaction forces vector is used as a initial guest for the reaction forces equations. For simplicity, during simulation we use direct linear equations resolution.

Definition of the equations in sections IV and V were conducted to provide optimal numerical procedures. Note that by optimal it should be understand as fast, but also numerically stable. Speed is obtained mainly by removing repetitions on term calculations. To obtain numerical stability, it is required to understand the geometric interpretation of the calculated terms. For stability, a accurate calculation of angular velocity is critical, where direct methods introduce complex terms that have always a zero value.

A. Definition of reaction forces equations

The calculation of Euler-Lagrange's equations on base of the energy functions with nonholonomic constrains in the form

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \lambda D^T(q,\dot{q}) \tag{35}$$

where λ are the reaction forces of the nonholonomic constrains given by

$$D(q,\dot{q})\dot{q} = 0. \tag{36}$$

Reaction forces calculation requires the transformation of equation (36) to the form (26)

$$D(q,\dot{q})\ddot{q} = b(q,\dot{q}). \tag{26}$$

Numerical procedures for the calculation of $M(q), C(q,\dot{q}), g(q)$ has been shown in section IV. Numerical procedures for the calculation of coeficients in equation (26) are described in section V-C.

To build linear equations for λ, substitute (35) in (26) to get

$$D(q,\dot{q})M^{-1}(q)\left(\lambda D^T(q,\dot{q}) - C(q,\dot{q})\dot{q} - g(q)\right) = b(q,\dot{q}) \tag{37}$$

The final reaction forces equations have the form

$$D(q,\dot{q})M^{-1}(q)\lambda D^T(q,\dot{q}) = b(q,\dot{q}) + D(q,\dot{q})M^{-1}(q)\left(C(q,\dot{q})\dot{q} + g(q)\right) \tag{38}$$

Resolution of (38) is realized online. Calculation of all matrix elements is realized numerically, and then the liniar equation is solved. Inverse of matrix $M(q)$ exists, because the Riemannian metric provide positive defined matrices for the considered mechanical systems.

VII. Example models

Presented equations are calculated in the Computer Algebra System Maxima. LaTeX representation is exported from maxima and directly introduced to the article content. There is not manual traduction of the equations.

The numerical procedures are also created from exported equations from Maxima to fortran code. Numerical procedures are written using the exported equations as black box components.

Rotation transformations are defined using the base

$$\Omega_x(a) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{pmatrix},$$
$$\Omega_y(a) := \begin{pmatrix} \cos a & 0 & -\sin a \\ 0 & 1 & 0 \\ \sin a & 0 & \cos a \end{pmatrix},$$
$$\Omega_z(a) := \begin{pmatrix} \cos a & -\sin a & 0 \\ \sin a & \cos a & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

A. The falling rolling disk

Following [1], the configuration space for the falling rolling disk is $Q = \mathbb{R}^3 \times S^1 \times S^1 \times S^1$ and it is parametrized by the coordinates $q = (x, y, z, \theta, \varphi, \psi)$ where:

- x, y, z - position coordinates of the center of the disk in the global reference frame,
- θ - angle between the plane of the disk and the vertical axis,
- φ - heading angle,
- ψ - roll angle.

The geometry graph definition is a single vertex root→disk. The transformation and rotation are

$$T = [x, y, z] \tag{39}$$

$$B = \Omega_z(\varphi) \cdot \Omega_x(\theta) \cdot \Omega_y(-\psi) \tag{40}$$

$$B = \Omega_z(\varphi)\Omega_x(\theta)\Omega_y(\psi) \tag{41}$$

Matrix $M(q)$ is diagonal and have values

$$\begin{pmatrix} m & 0 & 0 & 0 & 0 & 0 \\ 0 & m & 0 & 0 & 0 & 0 \\ 0 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & I_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin^2\psi\, I_2 + \cos^2\psi\, I_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sin^2\theta\, I_3 + \left(1 - \sin^2\psi\right)\cos^2\theta\, I_2 + \sin^2\psi\,\cos^2\theta\, I_1 \end{pmatrix}$$

Matrix $C(q, \dot{q})$ is not zero in the positions

$$\begin{aligned}
C(q,\dot{q})[4][5] &= \cos\psi\,\sin\psi\,\dot{y}\,I_1 - \cos\psi\,\sin\psi\,\dot{y}\,I_2 \\
C(q,\dot{q})[4][6] &= \cos\psi\,\sin\psi\,\cos^2\theta\,\dot{z}\,I_2 - \cos\psi\,\sin\psi\,\cos^2\theta\,\dot{z}\,I_1 \\
C(q,\dot{q})[5][4] &= \cos\psi\,\sin\psi\,\dot{y}\,I_2 - \cos\psi\,\sin\psi\,\dot{y}\,I_1 \\
C(q,\dot{q})[5][6] &= -\cos\theta\,\sin\theta\,\dot{z}\,I_3 + \left(1 - \sin^2\psi\right)\cos\theta\,\sin\theta\,\dot{z}\,I_2 + \sin^2\psi\,\cos\theta\,\sin\theta\,\dot{z}\,I_1 \\
C(q,\dot{q})[6][4] &= \cos\psi\,\sin\psi\,\cos^2\theta\,\dot{z}\,I_1 - \cos\psi\,\sin\psi\,\cos^2\theta\,\dot{z}\,I_2 \\
C(q,\dot{q})[6][5] &= \cos\theta\,\sin\theta\,\dot{z}\,I_3 + \left(\sin^2\psi - 1\right)\cos\theta\,\sin\theta\,\dot{z}\,I_2 - \sin^2\psi\,\cos\theta\,\sin\theta\,\dot{z}\,I_1
\end{aligned}$$

For the disk model, it is possible to calculate the nonholonomic constraints directly. This is because the coordinate θ match the angle between the normal vector to the ground and the normal vector to the disk plane. This means that the normalization factor in (25) is $\|\cos(\theta)\|$. Calculations of the constraint equations confirm this relation.

Equation (2) can be calculated directly to be

$$\begin{aligned}
(-\sin\theta\,\cos\varphi\,\dot{z} - \cos\theta\,\dot{y}\,\sin\varphi - \dot{x}\,\cos\varphi)\,r + \dot{\psi} &= 0 \\
(-\sin\theta\,\sin\varphi\,\dot{z} - \dot{x}\,\sin\varphi + \cos\theta\,\dot{y}\,\cos\varphi)\,r + \dot{\theta} &= 0 \\
\sin\theta\,\dot{y}\,r + \dot{\varphi} &= 0
\end{aligned}$$

B. Bike model

A local coordinate system fixed to the frame of the bicycle is selected, its coordinates on the global coordinate system are given by coordinates $(x, y, z, \tau_x, \tau_y, \tau_z)$.

Bicycle geometry is defined by fixed parameters

$$(\alpha_1, \alpha_2, d, l_1, l_2, r)$$

and state space coordinates

$$q = (x, y, z, \tau_x, \tau_y, \tau_z, \beta, \varphi_1, \varphi_2).$$

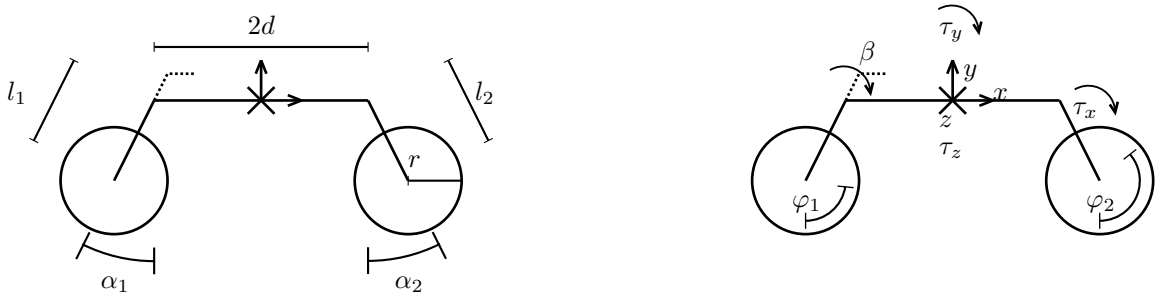
The bicycle model geometry graph is provided in figure 1. The transformations and rotations matrices are as follows:

$$T_g = (x, y, z)^T \quad (42)$$
$$T_l = (-d - l_1 \sin(\alpha_1), -l_1 \cos(\alpha_1), 0)^T \quad (43)$$
$$T_r = (d + l_2 \sin(\alpha_2), -l_2 \cos(\alpha_2), 0)^T \quad (44)$$

$$B_g = \Omega_x(\tau_x)\,\Omega_y(\tau_y)\,\Omega_z(\tau_z) \quad (45)$$
$$B_l = \Omega_z(\alpha_1)\,\Omega_y(\beta)\,\Omega_z(-\alpha_1)\,\Omega_z(\varphi_1) \quad (46)$$
$$B_r = \Omega_z(\varphi_2) \quad (47)$$

Note that this definition of the bicycle model is complete and unambiguous. All calculations are realized in base of the equations provided.

To ilustrate the model, figure 3 is provided.



(a) Side view - geometry parameters

(b) Side view - state variables

Fig. 3: Bike coordinates

Equations for the bicycle model create to large equations to be displayed in this format.

VIII. Conclusions

A systematic procedure for generation of symbolic and numerical models for mechanical system is presented. At the numerical level, procedures are created to calculate the Euler-Lagrange equations and the reaction forces of nonholonomic constraint equations.

Output equations are well knows for simple models like the rolling falling disk. For more complex models like a bicycle, there are not available models in the literature that could serve as a reference for comparation. Possible errors during the creation of the model equations are possible only on the model geometry definition. The geometric definition model provided as a acyclic directed graph, is easy to validated and containt the minimal set of information to uniquely define the model.

The final validation of the created model is expected to be implementation of control algorithms on real mechanical models.

ACKNOWLEDGMENT

This work was supported by the Polish National Science Center (NCN) under the Grant No.: 2011/01/N/ST8/01894.

References

[1] A. Bloch, *Nonholonomic mechanics and control*, Vol. 24, Springer Verlag, 2003.
[2] J Cortés, S Martinez, JP Ostrowski, and H Zhang, *Simple mechanical control systems with constraints and symmetry*, SIAM Journal on Control and Optimization **41** (2002), no. 3, 851–874.
[3] David EH Jones, *The stability of the bicycle*, Physics today **59** (2006), no. 9, 51.
[4] JDG Kooijman, JP Meijaard, Jim M Papadopoulos, Andy Ruina, and AL Schwab, *A bicycle can be self-stable without gyroscopic or caster effects*, Science **332** (2011), no. 6027, 339–342.
[5] JDG Kooijman, AL Schwab, and JP Meijaard, *Experimental validation of a model of an uncontrolled bicycle*, Multibody System Dynamics **19** (2008), no. 1-2, 115–132.

[6] David JN Limebeer and Robin S Sharp, *Bicycles, motorcycles, and models*, Control Systems, IEEE **26** (2006), no. 5, 34–61.
[7] FG Maunsell, *Why does a bicycle keep upright?*, The Mathematical Gazette (1946), 195–199.
[8] JP Meijaard, Jim M Papadopoulos, Andy Ruina, and AL Schwab, *Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review* (2007).
[9] JP Meijaard and AL Schwab, *Linearized equations for an extended bicycle model*, Iii european conference on computational mechanics, 2006, pp. 772–772.
[10] Clementina D Mladenova, *Applications of lie group theory to the modeling and control of multibody systems*, Multibody System Dynamics **3** (1999), no. 4, 367–380.
[11] P Sanjuan Szklarz and E Jarzębowska, *Coordinate-free formulation of nonholonomic constraints for wheeled robots*, Applied Non-Linear Dynamical Systems **1** (2014), no. 1.
[12] Arend L Schwab, Jaap P Meijaard, and Jim M Papadopoulos, *Benchmark results on the linearized equations of motion of an uncontrolled bicycle*, Journal of mechanical science and technology **19** (2005), no. 1, 292–304.
[13] Archibald Sharp, *Bicycles & tricycles: a classic treatise on their design and construction*, Courier Dover Publications, 2013.
[14] Robin S Sharp, *On the stability and control of the bicycle*, Applied mechanics reviews **61** (2008), no. 6, 060803.

