VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

MATEMATICKÉ MODELY ŘÍZENÍ ROBOTICKÝCH MECHANISMŮ

MATHEMATICAL MODELS OF ROBOTIC MECHANISMS CONTROL

HABILITAČNÍ PRÁCE
HABILITATION THESIS

AUTOR
AUTHOR

PETR VAŠÍK

BRNO 2016

Contents

1 Introduction

The history of robotics in the modern sense begins after the World War II by the development of remotely controlled mechanical manipulators of master–slave type. Shortly afterwards, in 1954, the master part was endowed by a programmable unit and attached to the Computer Numerically Controlled (CNC) machine tool for accurate milling of low–volume, high–performance aircraft parts. In 1961, the first programmable robot manipulator was installed in a General Motors plant. Since then the progress is divided into various directions, e.g. positioning accuracy improvement for CNCs, computational schemes for the gravity and Coriolis force control method in the 1980s or modifiability for different assembly operations. Nevertheless, these industrial manipulators started the development of advanced tools for the kinematics and dynamics description.

In the 70s, Hirose, [19], described a robotic system similar to the manipulators by construction, i.e. the robot composed of links and motorised joints. But unlike the manipulators, the robotic snake was unearthed and therefore moving robot and thus to the kinematic description certain nonholonomic conditions on the friction forces and velocity vectors were added leading to the system of differential equations and transforming the problem into the field of mathematical control theory. Indeed, the resulting system is a dynamical system with input functions that provide the prescribed motion by controlling the joint actuators. Mathematically, the foundations of the control theory were laid by Pierre–Simon Laplace, and it was further elaborated by James Clerk Maxwell in 1868 from the point of view of the oscillation theory and by W. Hurewitz in 1947 who analysed the stability conditions.

Within this text we rather use so–called modern control theory meaning that the controlling dynamical system is expressed in a matrix form $\dot{q} = A \cdot u$, where q denotes the point of a phase space, A is a controlling matrix and u is the input vector. The columns A_i af A then represent the vector fields that describe the direction of an infinitesimal motion w.r.t. the appropriate i–th element of u and thus a motion direction from a given point within the phase space is obtained as the linear combination of A_i. This approach is suitable for modelling and avoids the non–linearity obstructions. If a phase space is considered as a manifold and the motion generators as the vector fields, the problem is translated into the language of differential geometry. Indeed, e.g. the controllability analysis is then reduced to the dimension conditions on the Lie algebra generated by A_i by means of a Lie bracket. Moreover, the Lie bracket generated vector fields play an important role in the spatial motion of a snake robot. Note that the analysis of the generating process also leads to the classification of the robotic mechanisms according to the filtration of the appropriate Lie algebra w.r.t. the number of Lie brackets used, e.g. 3 link robotic snake's Lie algebra filtration is (2,3,5), i.e. there are two controlling vector fields, one their bracket and two additional independent Lie brackets of order two. Based on this classification, Ishikawa [27] introduced a mechanism of filtration (3,6) called a trident snake robot. We handle this mechanism within Section 4.1.

Consequently, the tools of sub–Riemannian geometry were used to adopt the con-

trolling system for the Lie bracket motions simulations in order to estimate the model error. This lead to the notion of nilpotent approximation, see Section 5.

Quite recently, the calculus of Conformal Geometric Algebra (CGA), see [15], was employed to the analytical description of robotic manipulators in order to decrease the computational complexity of position calculations.

In Section 2, we mention a precise definition of some elementary notions of differential geometry such as curves, manifolds, tangent space etc. Using these notions in Section 3, we provide an illustrating example of geometric control theory on a simple car model and mention an exact link between a dynamical system and differential geometry. In Section 4 we recall the kinematic description of the robotic snakes, including the trident snake and a 3–link robotic snake. Section 5 is devoted to the so–called nilpotent approximation of the controlling distribution. We describe the algorithm in a more detailed way compared to [43], Appendix 2. A global control model based on a serpenoid curve is briefly mentioned in Section 6. Next, we resume some facts about the Conformal Geometric Algebra in Section 7 and describe the translation of the snake robots' kinematics into the language of CGA. The author's results in this field and further details of Sections 3–7 can be found in [53, 43, 22, 23, 24] as Appendices 1–5, the results about the differential kinematics described in Section 7.2 can be found in [21]. Finally, we show another directions of author's research in differential geometry within the last Section 8.

2 Elementary notions of differential geometry

In this section we summarize the elementary notions of differential geometry that appear within this text. All definitions in this Section are rather classical and can be found in textbooks of differential geometry, see e.g. [4, 32, 55].

2.1 Curves

There exist two elementary approaches to the notion of a curve. In geometry, we usually understand a curve to be a particular set of points within a plane (or generally within n-dimensional Euclidean space $\mathbb{E}_n$). On the other hand, in mathematical analysis a curve often represents a graph of a smooth function which describes the trajectory of a point motion.

This „dynamical“ approach can be specified as follows: consider an open interval $I \subset \mathbb{R}$ whose elements represent time values. A mapping $f : I \to \mathbb{E}_n$ is called a motion in $\mathbb{E}_n$. Instead of motion we sometimes use path. If a basis is chosen in $\mathbb{E}_n$ then $f(t) = (f^1(t), \dots, f^n(t))$ is an n-tuple of real functions and thus the motion is identified with a vector valued function $f : I \to \mathbb{R}^n$. The derivative of such function is then called the motion velocity. We say that the motion $f(t)$ is of class C^r if all its components $f^i(t)$ share common derivatives up to order r. Clearly, the above definition does not depend on the choice of the coordinate system. Next, we say that a motion f is regular if $f'(t) \neq \vec{0}$ for all $t \in I$. This means that a regular motion has non zero

velocity at any point. Motion f is called simple if the condition $t_1 \neq t_2 \Rightarrow f(t_1) \neq f(t_2)$ holds. This means that the trajectory does not contain any self intersections.

Definition 2.1. A set $\mathcal{C} \subset \mathbb{E}_n$ is called a simple curve of class C^r if there exists such a simple regular motion $f : I \to \mathbb{E}_n$ of class C^r that $\mathcal{C} = f(I)$. A mapping f is then called a parametrization of a simple curve $f(I)$.

Quite analogously, one can define a surface $S \subset \mathbb{E}_3$, see e.g. [4].

According to the following proposition, any two parametrizations of the same simple curve differ by reparametrization.

Proposition 2.2. *Mappings $f(t) : I \to \mathbb{E}_n$ and $g(\tau) : J \to \mathbb{E}_n$ are parametrizations of the same simple curve $\mathcal{C}$ of class C^r if and only if there exists a bijection $\varphi : J \to I$, $t = \varphi(\tau)$ of class C^r such that for all $\tau \in J$ the assertion $\frac{d\varphi}{d\tau} \neq 0$ holds and $g(\tau) = f(\varphi(\tau))$. Function φ is then called the reparametrization or a parameter transformation of a curve $\mathcal{C}$.*

Definition 2.3. The set $\mathcal{C} \subset \mathbb{E}_n$ is called a curve of class C^r if for any point $p \in \mathcal{C}$ there exists a neighbourhood U_p in $\mathbb{E}_n$ such that $\mathcal{C} \cap U_p$ is a simple curve of class C^r. The parametrization of the intersections $\mathcal{C} \cap U_p$ is called a local parametrization of a curve $\mathcal{C}$.

If $f : I \to \mathbb{E}_n$ is some local parametrization of a curve $\mathcal{C}$ then the line determined by the point $f(t_0) \in \mathcal{C}$ and a vector $f'(t_0)$ is called the tangent line of a curve $\mathcal{C}$ at a point $f(t_0)$.

Definition 2.4. We say that two curves $\mathcal{C}, \overline{\mathcal{C}} \subset \mathbb{E}_n$ at a common point $p \in \mathcal{C} \cap \overline{\mathcal{C}}$ have the contact of order k if there exist their local parametrizations $f(t)$ and $\overline{f}(t)$, $f(t_0) = \overline{f}(t_0) = p$, such that $\frac{d^i f(t_0)}{dt^i} = \frac{d^i \overline{f}(t_0)}{dt^i}$ for all $i = 1, \ldots, k$.

Clearly, two curves have the contact of order 1 at a common point if and only if they share the common tangent line at that point. It is also true that the tangent line is the only line having the contact of order 1 with a curve at a particular point.

2.2 Manifolds

A crucial notion of differential geometry is that of a manifold. It can be understood as a generalization of the notions of curves and surfaces. For better understanding we employ the idea of the Earth cartography. Indeed, a sphere can not be described completely by one planar chart but it can be glued from a set of charts ordered into an atlas. Strictly speaking, any point of a sphere has a neighbourhood such that it can be uniquely mapped on a subset of a plane. We use the notions of charts and atlas even in the following text, where we define a manifold precisely. First, let us recall that a topological space is called Hausdorff if its any two points can be separated by two open sets. Next we recall that a homeomorphism is a continuous bijective mapping whose inverse is also continuous.

An n–dimensional topological manifold is a Hausdorff space M with a countable basis which is locally homeomorphic to $\mathbb{R}^n$, i.e. for every point $x \in M$ there exists its open neighbourhood $U \subset M$ and a homeomorphism $\varphi : U \to \varphi(U) \subset \mathbb{R}^n$. The pair (U, φ) is called a local chart. A system of charts $(U_\alpha, \varphi_\alpha), \alpha \in I$ on M such that U_α cover whole M is called an atlas. The demand on the countable basis guarantees that a finite or countable system of charts covering whole M can be chosen.

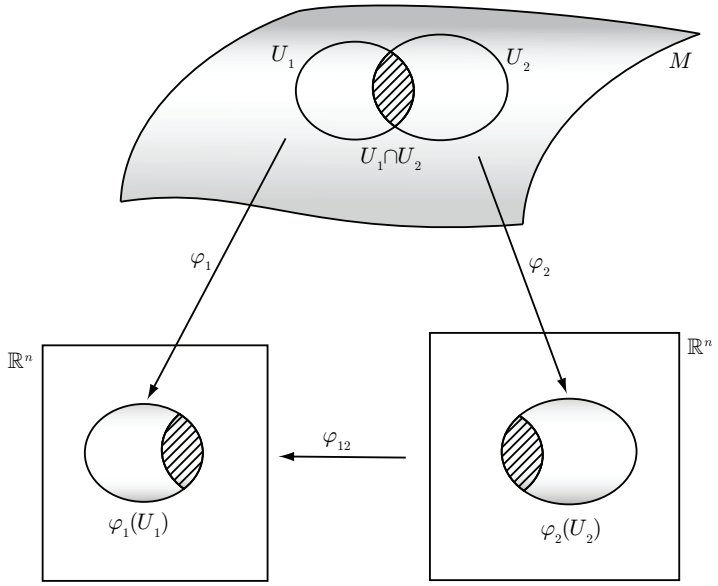


Figure 1: Manifold charts

Two local charts (U_1, φ_1), (U_2, φ_2) induce a mapping $\varphi_{12} := \varphi_1 \circ \varphi_2^{-1} : \varphi_2(U_1 \cap U_2) \to \varphi_1(U_1 \cap U_2)$ between two subsets of $\mathbb{R}^n$ called the chart changing mapping, see Figure 1. We say that an atlas $(U_\alpha, \varphi_\alpha)$ of a manifold M is of class C^k, if all chart changing mappings $\varphi_{\alpha\beta}$ are diffeomorphisms of class C^k (i.e. $\varphi_{\alpha\beta}$ is a bijection of class C^k such that its inverse is also of class C^k). A chart (U_0, φ_0) is called compatible with the atlas $(U_\alpha, \varphi_\alpha)$ of class C^k if any chart changing mapping $\varphi_{0\alpha}$ is a diffeomorphism of class C^k. Atlas $(U_\alpha, \varphi_\alpha)$ of class C^k is called complete if it contains all compatible charts.

Definition 2.5. A differentiable manifold of class C^k is a topological manifold M with a complete atlas of class C^k.

A mapping φ from a local chart (U, φ) is given by an n–tuple of functions $(\varphi^1, \ldots, \varphi^n)$ denoted by $(x^1, \ldots, x^n)$ or (x^i) that are called local coordinates of a manifold M, the set U is called a coordinate neighbourhood. We say that a mapping $f : M \to N$ between two manifolds is of class C^k if for any $x \in M$ and any chart (V, ψ) on N such that $f(x) \in V$, there exists a chart (U, φ) on M such that $x \in U$ and the mapping $\psi \circ f \circ \varphi^{-1}$ is of class C^k. A mapping $\psi \circ f \circ \varphi^{-1}$ is called a coordinate form of mapping f. If (y^p) are local coordinates on N, this coordinate form is $y^p = f^p(x^1, \ldots, x^n)$. Analogically, we define a function $f : M \to \mathbb{R}$ of class C^k. In the sequel, we assume

all manifolds (mappings, functions) to be of class C^∞ and will be called smooth. All smooth manifolds and their mappings form a category denoted as $\mathcal{M}f$.

2.3 Tangent bundle

We recall that tangent vectors of a surface $\mathcal{S} \subset \mathbb{E}_3$ at a point $x \in S$ are defined as the tangent vectors of such curves on $\mathcal{S}$ that contain the point x. We use this idea to establish the notion of a tangent vector to a manifold M. A smooth mapping $f : I \to M$ is called a path on a manifold M. In the sequel and w.l.o.g., we assume that the interval I contains zero.

Definition 2.6. We say that two paths $f, g : I \to M$ satisfying $f(0) = g(0) = a$ have a contact at a point $a \in M$ if there exists a coordinate neighbourhood U of a point a with local coordinates (x^i) such that

$$\frac{d(x^i \circ f)(0)}{dt} = \frac{d(x^i \circ g)(0)}{dt}.$$

Clearly, the above definition is independent on the choice of local coordinates. The equivalence class of paths $f(t)$ on M satisfying $f(0) = a$ and having a contact at $a \in M$ is called a tangent vector of a manifold M at a point a and we denote it as $t_a = \frac{df(0)}{dt}$. The elements $\xi^i := \frac{d(x^i \circ f)(0)}{dt}$ are then called the coordinates of a vector t_a in local coordinates (x^i). If $\varphi = \varphi(x^i) : M \to \mathbb{R}$ is an arbitrary smooth function on U then

$$t_a\varphi := \frac{d(\varphi \circ f)(0)}{dt} = \sum_{i=1}^{n} \frac{\partial\varphi(a)}{\partial x^i} \frac{d(x^i \circ f)(0)}{dt} = \sum_{i=1}^{n} \frac{\partial\varphi(a)}{\partial x^i} \xi^i. \tag{2.1}$$

The value $t_a\varphi$ defined by (2.1) is called a derivative of a function φ in the direction of vector t_a. Clearly, for arbitrary smooth functions $\varphi, \psi : M \to \mathbb{R}$ defined on the neighbourhood of a point a the following holds:

$$t_a(r\varphi + s\psi) = rt_a\varphi + st_a\psi, \quad t_a(\varphi \cdot \psi) = \varphi(a) \cdot t_a\psi + \psi(a) \cdot t_a\varphi, \quad r, s \in \mathbb{R}. \tag{2.2}$$

Consequently, it is possible to define a tangent vector as an operator t_a which assigns a real number $t_a\varphi$ to a function $\varphi : M \to \mathbb{R}$ and which satisfies (2.2). An example of a tangent vector is an operator $\left(\frac{\partial}{\partial x^i}\right)_a$ which to a function φ assigns its derivative w.r.t. x^i at a point a.

The set T_aM of all tangent vectors of a manifold M at a point a is called the tangent space of M at a. Then T_aM is n–dimensional vector space with the basis formed by tangent vectors $\left(\frac{\partial}{\partial x^i}\right)_a$. The disjoint union of tangent spaces at every a is denoted by $TM := \cup_{a \in M} T_aM$. Clearly, TM is $2n$–dimensional differentiable manifold called the tangent space. Together with a natural projection $p : TM \to M$ which to a tangent vector $t_a \in T_aM$ assigns the contact point a, we have a tangent bundle denoted by $TM \to M$.

2.4 Vector fields

Definition 2.7. Let $TM \to M$ be the tangent bundle of a manifold M. By a vector field on M we understand a smooth mapping $X : M \to TM$ which to any point $a \in M$ assigns a tangent vector $X(a) \in T_aM$.

If (x^i) are local coordinates on the neighbourhood U of a point $x \in M$ then the vector field X can be expressed in the form

$$X = \sum_{i=1}^{n} X^i \frac{\partial}{\partial x^i},$$

where $X^i = X^i(x)$ are smooth functions defined on U. For any smooth function $f : M \to \mathbb{R}$, it is possible to define its derivative $Xf : M \to \mathbb{R}$ along the vector field X by $(Xf)(a) = X(a)f$ where the right hand side stands for a derivative in the direction of a vector $X(a) \in T_aM$ as in (2.2). In local coordinates clearly $Xf = \sum_{i=1}^{n} X^i(x)\frac{\partial f}{\partial xi}$. It turns out that the set $\chi(M)$ of all vector fields on M can be identified with the space of all derivatives of the algebra of smooth functions $C^\infty(M, \mathbb{R})$, i.e. with $\mathbb{R}$–linear operators $D : C^\infty(M, \mathbb{R}) \to C^\infty(M, \mathbb{R})$ satisfying $D(fg) = D(f)g + fD(g)$.

A path $f : I \to M$ is called an integral curve of a vector field X if the vector $X(f(t))$ is tangent to f at $f(t)$ for any $t \in I$, i.e.

$$\frac{df(t)}{dt} = X(f(t)) \quad \forall t \in I.$$

Any integral curve is thus a solution of a system of ODEs $\frac{dx^i}{dt} = X^i(x^1, \ldots, x^n)$. From the theory of ODEs it follows that for any point $x \in M$ there exists an open interval I_x containing 0 and an integral curve $f_x : I_x \to M$ of a vector field X such that $f_x(0) = x$. If the interval I_x is maximal then f_x is unique. Furthermore, the set $\mathcal{D}X := \cup_{x \in M} I_x \times \{x\} \subset \mathbb{R} \times M$ is open and a mapping $\mathrm{Fl}^X : \mathcal{D}X \to M$ defined by $\mathrm{Fl}^X(t, x) = f_x(t)$ is smooth. The mapping Fl^X is called the flow of a vector field X.

For any pair of vector fields X, Y on M there exists a unique vector field $[X, Y]$ on M such that for any function f on M the assertion $[X, Y]f = X(Yf) - Y(Xf)$ holds. The vector field $[X, Y]$ is called a Lie bracket of vector fields X, Y. If $X^i(x)$ and $Y^i(x)$ are the coordinate forms of X and Y then the Lie bracket $[X, Y]$ is of the form $\sum_{j=1}^{n} X^j \frac{\partial Y^i}{\partial x^j} - Y^j \frac{\partial X^i}{\partial x^j}$. Note that the set of all vector fields $\chi(M)$ on a manifold M together with the Lie bracket form a Lie algebra which, generally, is defined as a vector space $\mathfrak{g}$ over a field F together with a binary, bilinear, anti–commutative operation denoted by $[\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ defined by $[X, Y] = XY - YX$ for $X, Y \in \mathfrak{g}$ assuming that an algebraic product is defined on $\mathfrak{g}$ and, furthermore, for any three elements $X, Y, Z \in \mathfrak{g}$ the Jacobi identity holds:

$$[X, [Y, Z]] + [Z, [X, Y]] + [Y, [Z, X]] = 0.$$

In the following text, we use some more advanced notions of differential geometry such as e.g. the Lie derivative, Lie group, fibred manifold and connection, the definition of which is omitted and we refer to e.g. [32, 55] for the details.

3 Motivation for geometric control

3.1 Local control and differential geometry

We now show the connection between the dynamical systems and vector fields, more precisely we show the simple car model description by both methods, for classical approach see e.g. [35]. Furthermore, on the geometrical background we discuss the model controllability and show the meaning of a Lie bracket.

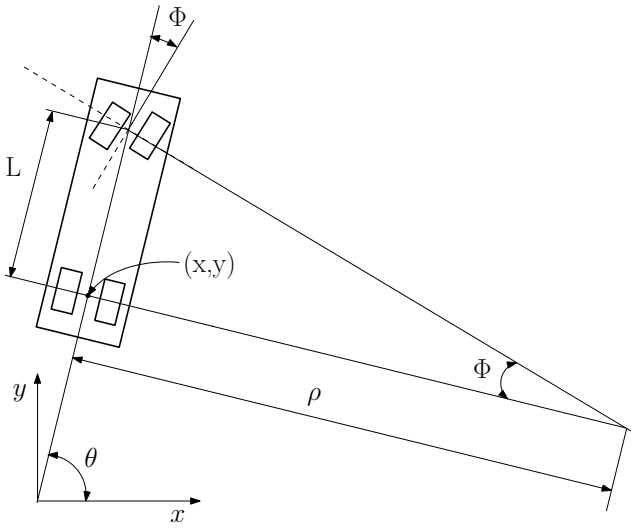


Figure 2: Simple car model

For the model description see Figure 2 and note that the fixed coordinate system (x, y) is attached. The car moves on a planar surface and thus the configuration space is isomorphic to $\mathbb{R}^2 \times \mathbb{S}^1 \times \mathbb{S}^1$ with coordinates denoted by (x, y, θ, Φ), first two describing the position, the third one the car orientation w.r.t. the fixed coordinate system and Φ determining the shape of the car, i.e. the front wheels orientation. The non–holonomic condition on the rear wheels reads that the velocity vector is always tangential to the trajectory of the rear axle midpoint (x, y). Mathematically, this condition can be expressed as

$$-\dot{x}\sin\theta + \dot{y}\cos\theta = 0,$$

which is the form of the Pfaffian equation, i.e. dually in the form $P\mathrm{d}x + Q\mathrm{d}y = 0$ for a vector field $(P(x, y), Q(x, y))$. This is satisfied if $\dot{x} = \alpha\cos\theta$ and $\dot{y} = \alpha\sin\theta, \alpha \in \mathbb{R}$, i.e. the α multiple of the velocity vector v of the car. If we set $\alpha = \| v \| =: s$, we obtain two configuration equations $\dot{x} = s\cos\theta$ and $\dot{y} = s\sin\theta$. To describe the car position and spatial orientation (not front wheels orientation) completely, we need to determine the equation for $\dot{\theta}$, too. First observe that $\rho = \frac{L}{\tan\Phi}$. If w denotes the distance travelled by the car then clearly $w = \frac{2\pi\rho}{2\pi}\theta = \rho\theta$ for a fixed Φ and thus $\mathrm{d}w = \rho\mathrm{d}\theta$. Consequently

$$\mathrm{d}\theta = \frac{\tan\Phi}{L}\mathrm{d}w.$$

Using the fact that $\dot{w} = s$ this yields

$$\dot{\theta} = \frac{s}{L} \tan \Phi.$$

To comply with the notation used in the field of dynamical systems we should specify the action variables (the input). In this model, the action variables are clearly the speed s controlled by the throttle, and Φ controlled by the steering wheel. As the inputs are denoted by (u_s, u_Φ), we obtain a control system

$$\begin{aligned} \dot{x} &= u_s \cos \theta, \\ \dot{y} &= u_s \sin \theta, \\ \dot{\theta} &= \frac{u_s}{L} \tan u_\Phi. \end{aligned} \tag{3.1}$$

In addition, further restriction such as the steering angle limitation must be applied. Furthermore, from the point of view of dynamical systems, note that the exact solution depends on the actual state of the input functions. Nevertheless, as the system serves for modelling, it is considered as the dynamic feedback control law and the solution of a linearized system is involved in each iteration.

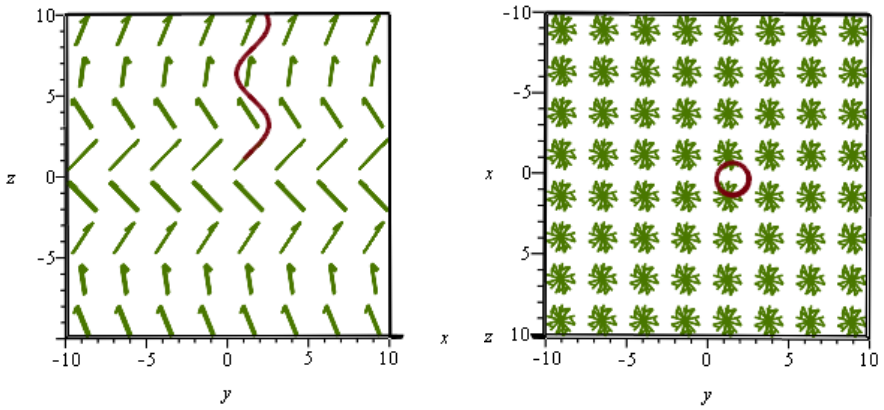
The question arises whether there exists a different way of the problem modelling that could be more convenient with some additional topics. This is where the modern geometric control theory comes to action. Indeed, if the configuration space is considered as a four dimensional manifold M with coordinates (x, y, θ, Φ), then the reconfiguration, i.e. both spatial, including the position and orientation, and shape (the front wheels orientation) transformations, are viewed as curves on the manifold M. Thus , equivalently, we may consider the evolution of the shape coordinate $\dot{\Phi}$ as an input function and obtain the modification of the system 3.1 in the form

$$\begin{aligned} \dot{x} &= u_s \cos \theta, \\ \dot{y} &= u_s \sin \theta, \\ \dot{\theta} &= \frac{u_s}{L} \tan \Phi, \\ \dot{\Phi} &= u_\Phi, \end{aligned} \tag{3.2}$$

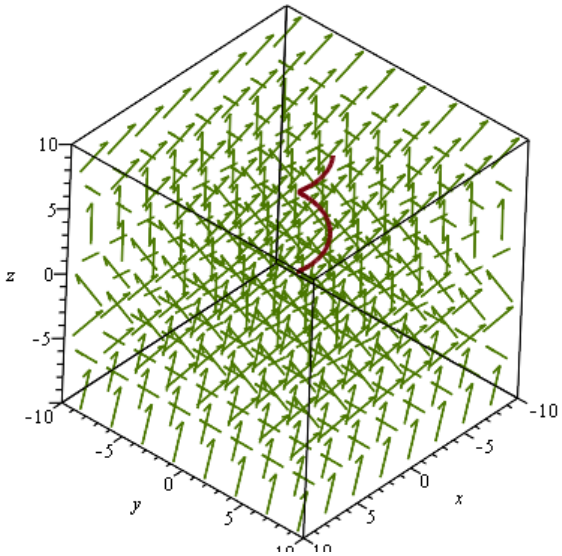
which is again a dynamical system, yet now in the Pfaffian form which defines the coordinates of the controlling vector fields explicitly as the input functions coefficients, i.e.

$$\begin{aligned} \text{steering: } S &= \frac{\partial}{\partial \Phi}, \\ \text{drive: } D &= \cos \theta \frac{\partial}{\partial x} + \sin \theta \frac{\partial}{\partial y} + \frac{\tan \Phi}{L} \frac{\partial}{\partial \theta}. \end{aligned}$$

The motion within the phase space is then realized along the integral curves of these vector fields, which are easily found for a fixed initial point, see Figure 3.



(a) θ evolution (b) x, y evolution



(c) Overall view

Figure 3: Integral curves

Clearly Figure 3(a) shows the periodic evolution of θ denoted as z within the graph, Figure 3(b) reads that the car with fixed steering angle rotates in x, y plane and Figure 3(c) shows the spatial form of the integral curve with fixed steering angle Φ. Very simple sequential motion planning algorithms then come to the reader's mind. One can either evaluate the vector fields S and D at the initial point x_0, move in the constant direction of their arbitrary linear combination into the point x_1, recalculate the vector fields again and proceed further by iteration, or determine the integral curve of the particular linear combination of vector fields S and D at x_0, move along the integral curve into the point x_1, recalculate the integral curve in x_1 and proceed by iteration, see Figure 4. Such algorithm demands sufficient computational capacity and suffers from its local character. Furthermore, the problem also lies in the question how to determine a curve that starts at x_0 and ends in a prescribed point x_1. Thus it is not

suitable for global motion planning, yet in the sequel we show its contribution to the robotic snake motion planning.

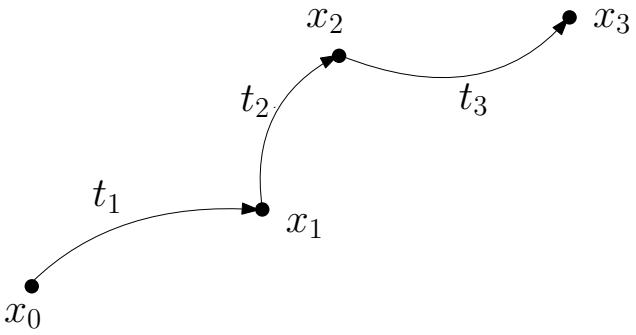


Figure 4: Motion planning

We described how a motion may be obtained by a linear combination of the controlling vector fields. Yet, there exists their more sophisticated combination called the Lie bracket. Note that we use the Lie bracket on the Lie algebra of vector fields. The calculation of a Lie bracket of vector fields X, Y on a manifold M with coordinates denoted by $\underline{x}$ at a point $x_0 \in M$ is quite simple, indeed

$$[X, Y](x_0) = \frac{\partial X}{\partial \underline{x}}(x_0)Y(x_0) - \frac{\partial Y}{\partial \underline{x}}(x_0)X(x_0),$$

where $\frac{\partial X}{\partial \underline{x}}$ stands for a Jacobi matrix of X.

To demonstrate the effect of a Lie bracket, let us consider the flows (or integral curves) of the appropriate vector fields. Then the Lie bracket measures the non–commutativity of the corresponding flows, see Figure 5.

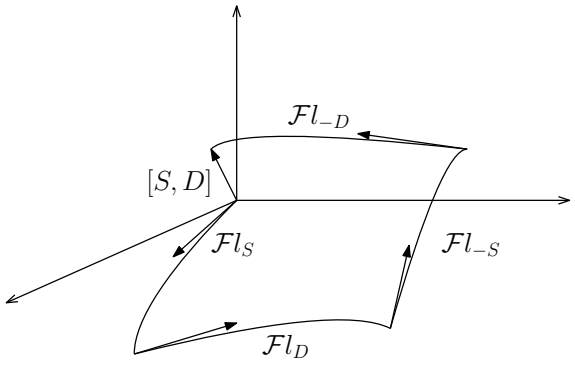


Figure 5: Lie bracket

In case of simple car model we obtain two more available motions, particularly

$$\text{rotation: } R = [S, D] = \frac{1}{L\cos^2\Phi}\frac{\partial}{\partial\theta},$$
$$\text{translation: } T = [R, D] = -\frac{\sin\theta}{L\cos^2\Phi}\frac{\partial}{\partial x} + \frac{\cos\theta}{L\cos^2\Phi}\frac{\partial}{\partial y}.$$

From the coordinate form it is clear that R corresponds to the change of the orientation only, i.e. it represents pure rotation, while T moves the car w.r.t. the spatial coordinates x, y only and thus it represents the translation. It is also clear that such motions cannot be realized directly in the motion planning algorithm but have to be formed as a combination of the controlling vector fields S and D. See Figure 6 for the motions interpretations. Note that e.g. the translation in the direction parallel to the rear axle is the answer to the parallel parking problem, see Figure 6(a).

(a) Translation (b) Rotation

Figure 6: Bracket motions

Clearly, parallel parking problem can be solved using the combination of S and D vector fields and, as shown above, the Lie bracket plays an important role in it. Furthermore, if the Lie bracket is understood as the composition of flows, see Figure 5, it becomes clearer that the Lie bracket motion is realized by so–called periodic input, see [42] or (4.5) for its exact form.

Finally, we discuss the application of the differential geometry approach to the solution of local controllability. It can be calculated directly that there are pointwise no linearly independent vectors generated by further iteration of the Lie bracket. Indeed, this means that the dimension of the Lie algebra generated by S, D, R and T is 4 so as the dimension of the configuration space, or more precisely of its tangent space, and thus, according to the Chow–Rashevski Theorem, the system is locally controllable. In other words, this means that it is possible to steer the system from a given initial state to any final state within its neighbourhood in finite time using the available controls.

On a simple car model we demonstrated the geometric approach to the control theory which is used in Appendices 2-5, [43, 22, 23, 24].

3.2 Further geometric concepts

Note that there are further notions of differential geometry that play an important role in the control theory. We mention two concepts that appear within this text.

Clearly, the control system (3.2) can be rewritten as

$$\dot{q} = u_s \cdot S(q) + u_\varphi \cdot D(q)$$

assuming that $q = (x, y, \theta, \Phi)$. Generally, any controlling system can be written as

$$\dot{q} = u_1 X_1(q) + \cdots + u_m X_m(q),\ q \in M,\ u = (u_1, ..., u_m) \in \mathbb{R}^m, \tag{3.3}$$

where X_i are smooth vector fields on M, see [29]. Indeed, the system (3.3) determines a family of vector spaces

$$\Delta(q) = \text{span}\{X_1(q), \ldots, X_m(q)\} \subset T_qM,\ q \in M$$

which, in case that the dimension of $\Delta(q)$ is constant for all $q \in M$, is called the distribution. We will focus on such cases only, furthermore, we suppose that the dimension of $\Delta(q)$ is lower than the dimension of T_qM and this defines so–called nonholonomic system. Consequently, it is possible to define a trajectory of (3.3) as a path $\gamma : [0,1] \to M$ for which there exists a function $u(\cdot) \in L^1([0,T], \mathbb{R}^m)$ such that γ is a solution of the ordinary differential equation

$$\dot{q}(t) = \sum_{i=1}^{m} u_i(t) X_i(q(t))$$

for almost every $t \in [0,T]$. Such a function $u(\cdot)$ is called a control associated with γ. Note that the existence of such path between two sufficiently close points is given by the local controllability. Now one can define a sub-Riemannian metric associated with (3.3) as a function $g : TM \to \mathbb{R} \cup \{\infty\}$ given by

$$g(q,v) = \inf \left\{ u_1^2 + \cdots + u_m^2 : \sum_{i=1}^{m} u_i X_i(q) = v \right\},$$

where the convention reads that $\inf \emptyset = +\infty$. With the notion of a metric we are ready to define the length of a path $\gamma(t)$ as

$$\text{length}(\gamma) = \int_0^T \sqrt{g(\gamma(t), \dot{\gamma}(t))} \mathrm{d}t$$

and finally the sub-Riemannian distance on M associated with (3.3) as

$$d(p,q) = \inf \text{ length}(\gamma)$$

for such γ that $\gamma(0) = p$ and $\gamma(T) = q$. Note that this concept allows measuring the trajectories (motions) of a mechanism defined by (3.3) assigning a finite measure to those trajectories whose tangent vectors are a linear combination of the controlling vector fields X_i. Infinite measure is assigned to the trajectories that are not solutions of (3.3) and thus the appropriate motion violates the mechanism's natural behaviour. In the context of a simple car model, this means that the rear wheels slip or slide and thus the velocity vector is no longer purely perpendicular to the rear axle.

In addition, we recall another important notion that refers to the Lie bracket generating process which is the filtration appropriate to the nonholonomic system. We use the same notation as in [29]. Let us by Δ^1 denote the distribution

$$\Delta^1 = \text{span}\{X_1, ..., X_m\}$$

and for $s \geq 1$ define

$$\Delta^{s+1} = \Delta^s + [\Delta^1, \Delta^s],$$

where $[\Delta^1, \Delta^s] = \text{span}\{[X, Y] : X \in \Delta^1, Y \in \Delta^s\}$. Then the filtration is understood as a sequence of dimensions of the distributions Δ^s, i.e. the filtration appropriate to the simple car model is (2,3,4) as the vector field T is the second order Lie bracket. Note that modelling of the Lie bracket motions leads to the concept of the nilpotent approximation that follows the sub–Riemannian geometry structure given by the dynamical system. Particular example will be given in Section 5.

Finally, geometrical concept of a connection plays an important role in the control theory, yet the exact use can vary and may demand quite deep knowledge of differential geometry. For the use of a connection in the control theory based on the kinematic energy optimization see e.g. [13, 46]. To find the correspondence between dynamical system, vector field and a connection we refer to [53], Appendix 1, where the connection is established by the projection onto the space of controlling parameters, canonical parameters and differential invariants of the controlling system. Indeed, if a solution of a dynamic system is considered as an integral curve (flow) of an appropriate vector field denoted by Y, then the canonical parameter s, defined by $\mathcal{L}_Y s = 1$, and the system of independent invariants I^k defined by $\mathcal{L}_Y I = 0$ of Y can be considered, where $\mathcal{L}_Y$ denotes the Lie derivative w.r.t. the vector field Y. Consequently, if we define a submersion

$$\varphi : \Delta \to (s, I^k),$$

then each fibre is formed by the family of integral surfaces which define a horizontal distribution Δ_h and therefore a connection. Thus the control is identified with a (torsion free) connection, see [53], Appendix 1 for further details, application to the dynamical system transformation and examples. For the differential geometrical background see [1, 50], for more details about dynamical systems see e.g. [44].

4 Kinematics of robotic snakes

Robotic snakes, or rather snake-like robots, are a good example of very sophisticated biologically inspired specimen. A snake, as a mobile biological unit, is very well adapted for an effective overcoming of obstacles and thus its terrain traversability is good in very complex environments [30]. From this point of view there are two very beneficial and fundamentally irreplaceable robotic applications. The first is the snake as an independent exploratory unit which is able to move in terrain that is hardly reachable or unreachable by means of the common mobile technology based on wheels or legs. One of many examples is a rescue mission in the earthquake affected areas [39, 40]. The snake robot could crawl through a destroyed building looking for people. The snake robot can also be used for surveillance and maintenance of complex and possibly dangerous structures such as nuclear plants and pipelines. Other application can be for example a snake-like robot used as a slender hyper-redundant manipulator, which is irreplaceable in industry and also in medical applications. Compared to the wheeled and legged mobile mechanisms, the snake robot offers high stability and superior traversability. Nowadays, snake-like robots are actively developing area in robotics, rising in popularity.

One of the most famous works has been done by Hirose and Umetami, in the early 1970's, they were the first to explore, design and develop limbless locomotors [19, 20, 41, 56]. They started the first bio-mechanical study using the real snakes and designed the first snake-like robot based on so–called serpentine locomotion. The first designs of Hirose's snake robots had modules with small passive wheels, and since then, most of the current developments by Downling (1997) [9], Chirikjian and Burdick (1990) [7], and Ostrowski (1995) [45] keep using the snake robots with wheels in order to facilitate forward propulsion.

Up to now, a variety of different snake-like robots has been designed, see [7, 31, 36, 47], some of which are currently used in practical tasks. It is important to note that there are other locomotion models using the principles of the biological snakes [6]: 1) Serpentine locomotion; 2) Rectilinear locomotion; 3) Concertina locomotion; 4) Side winding locomotion. The precise modelling of any of these motion components is quite difficult task [37]. There is a significant number of papers focused on searching for new snake–like mechanism constructions in order to realize one or more of the above models, [11, 38]. For the snake–like motion mimicking, basically three approaches are used (simplified classification).

First, the original Hirose approach, [19, 20], i.e. the representation of the kinematic model based on the serpenoid curve. This is well developed in many directions, [2]. Two main planar models, the serpentine locomotion (horizontal locomotion) and the travelling wave locomotion (vertical locomotion) are often adopted in the snake robot [5]. Furthermore, many authors present the optimized versions of this approach based on various parameter estimations. This is also often used for the verification of other solutions. We present a short description of serpenoid curve–based model within Section 6.

Second, heuristic and artificial intelligence approach based on so–called CPG (Central Pattern Generators) or other types of non–linear models. CPG produces so called rhythmic outputs defining a periodic motion of a robotic snake. It is realized as the intra-spinal artificial neural network. It is one of the mechanisms for neural control of locomotion in higher animals and humans [49]. In the field of modular robots, the CPG plays an important role in the motion description [8]. Different variations of this approach are still widely studied and developed [26].

Third, the analytical mechanic model based on the Newton-Euler formulation and Lagrangian [54, 42]. The fact that the energy of the system and the friction forces acting on the system are invariant with respect to the position and orientation of the snake robot is exploited to simplify the mathematical model. In the position calculation, the classical Euler angles can be used. The same calculations with much lower computational complexity are provided when the representations by quaternions or by the elements of so–called Conformal Geometric Algebra (CGA) [3, 16, 17] are employed. This approach is elaborated on a snake robot within Appendices 3–4, [22, 23], for trident snake robot see Appendix 5, [24].

Note that the above approaches differ not only by mathematical methods but also by suitability for particular applications. The division into global and local control is

vital. While heuristic methods suit well to the global control, the analytical approach solves rather local control, i.e. the way of actual movements and thus it can be used e.g in the situations where not enough joints can be operated, such as narrow spaces or difficult terrain due to obstacles.

In the sequel, we describe two particular mechanisms, i.e. trident snake and snake robot. We apply the geometric control methods and show elementary local motion planning algorithms based on the controlling vector fields. We shall stress that the Lie bracket motions are crucial to provide continuous lateral undulation and show the exact form of a periodic input that models them satisfactorily. Let us note that according to [42], the accuracy of modelling the bracket motions by periodic input up to order two is checked by so–called nilpotent approximation of the controlling distribution. We describe the whole transformation process for the 3–link snake, where the full strength algorithm must be used, in [43], Appendix 2. Within this text we compute the exact form of the trident snake's nilpotent approximation with comments and more details.

4.1 Trident snake robot

The mechanism of the trident snake robot was introduced in [27]. It consists of a body in the shape of an equilateral triangle with circumscribed circle of radius r and three rigid links (also called legs) of constant length l connected to the vertices of the triangular body by three motorised joints. In this text, we consider $r = 1$ and $l = 1$. To each free link end, a pair of passive wheels is attached to provide an important snake-like property that the ground friction in the direction perpendicular to the link is considerably higher than the friction of a simple forward move. In particular, this prevents slipping sideways. To describe the actual position of a trident snake robot we need the set of 6 generalized coordinates

$$q = (x, y, \theta, \Phi_1, \Phi_2, \Phi_3) =: (x_1, x_2, x_3, x_4, x_5, x_6)$$

as shown in Figure 7. Hence the configuration space is (a subspace of) $\mathbb{R}^2 \times \mathbb{S}^1 \times (\mathbb{S}^1)^3$. Note that a fixed coordinate system (x, y) is attached.

Let us now recall the kinematics description of the trident snake robot according to [27]. Let $x_i, y_i,\ i = 1, 2, 3$, denote the wheel positions, θ the absolute orientation and x, y the platform centre coordinates. Then

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x + \cos(\alpha_i + \theta) + \cos(\alpha_i + \theta + \Phi_i) \\ y + \sin(\alpha_i + \theta) + \sin(\alpha_i + \theta + \Phi_i) \end{pmatrix}, \tag{4.1}$$

where α_i denotes the i–th central angle within the platform triangle. Note that the parametrizations can vary by setting the angles within the triangular platform either $0, \frac{2\pi}{3}$ and $\frac{4\pi}{3}$ or $0, \frac{2\pi}{3}$ and $-\frac{2\pi}{3}$, etc. The non–slip and non–slide assumption on the wheels imply the following three nonholonomic constraints:

$$\dot{x}_i \sin(\alpha_i + \theta + \Phi_i) = \dot{y}_i \cos(\alpha_i + \theta + \Phi_i),\ i = 1, 2, 3. \tag{4.2}$$

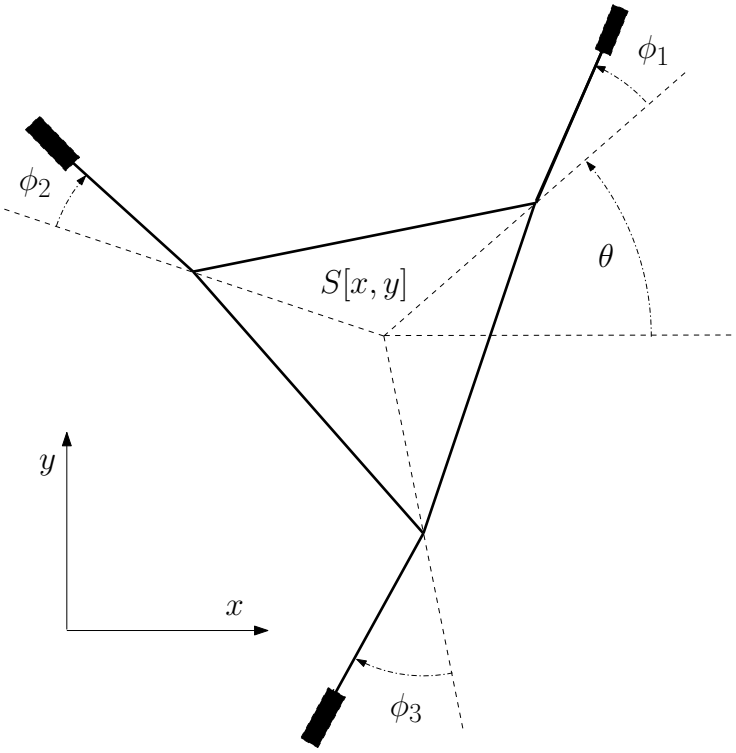


Figure 7: Trident snake robot model

Differentiating the equations (4.1) and substituting into (4.2) we obtain the controlling dynamical system. Similarly to the system (3.2) we transform it into the appropriate Pfaff system of ODEs which describes the local controllability. If $\dot{q}$ is expressed we obtain a control system $\dot{q} = G\mu$, more precisely

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\Phi}_1 \\ \dot{\Phi}_2 \\ \dot{\Phi}_3 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \\ \sin(\theta+\alpha_1+\Phi_1) & -\cos(\theta+\alpha_1+\Phi_1) & -1-\cos(\Phi_1) \\ \sin(\theta+\alpha_2+\Phi_2) & -\cos(\theta+\alpha_2+\Phi_2) & -1-\cos(\Phi_2) \\ \sin(\theta+\alpha_3+\Phi_3) & -\cos(\theta+\alpha_3+\Phi_3) & -1-\cos(\Phi_3) \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

where the vector (u_1, u_2, u_3) is the vector of controlling parameters and the control matrix G is a 6×3 matrix whose columns are considered as the controlling vector fields g_1, g_2, g_3.

It is easy to check that in regular points, i.e. such points where the matrix G is regular, these vector fields define a (bracket generating) distribution with growth vector $(3, 6)$. It means that in each regular point the vector fields g_1, g_2, g_3 together with their Lie brackets span the whole tangent space. Consequently, the system is controllable by Chow–Rashevsky theorem.

For sake of simplicity and according to [43], Appendix 2, we use the simplified vector fields where the rotation matrix is factored out which corresponds to the non–inertial coordinate system connected to the platform. The Lie algebra generating vector fields

g_1, g_2, g_3 with the parametrization $\alpha_1 = -\frac{2\pi}{3}, \alpha_2 = 0$ and $\alpha_3 = \frac{2\pi}{3}$ are then transformed as follows:

$$\begin{aligned} g_1 &= \partial_x + \sin(\Phi_1 - \tfrac{2\pi}{3})\partial_{\Phi_1} + \sin(\Phi_2)\partial_{\Phi_2} + \sin(\Phi_3 + \tfrac{2\pi}{3})\partial_{\Phi_3}, \\ g_2 &= \partial_y - \cos(\Phi_1 - \tfrac{2\pi}{3})\partial_{\Phi_1} - \cos(\Phi_2)\partial_{\Phi_2} - \cos(\Phi_3 + \tfrac{2\pi}{3})\partial_{\Phi_3}, \\ g_3 &= \partial_\theta - (1 + \cos(\Phi_1))\partial_{\Phi_1} - (1 + \cos(\Phi_2))\partial_{\Phi_2} - (1 + \cos(\Phi_3))\partial_{\Phi_3}. \end{aligned} \tag{4.3}$$

Similarly to the simple car model we shall explore the Lie bracket motions more precisely. Thus, we calculate the vector fields given by the Lie brackets of g_1, g_2, g_3 evaluated at 0 and denote them by $g_4 = [g_1, g_2], g_5 = [g_2, g_3]$ and $g_6 = [g_1, g_3]$. Their coordinate form with respect to the rotation free system (i.e. the non–inertial system connected with the platform) is the following:

$$\begin{aligned} g_4 &= \partial\Phi_1 + \partial\Phi_2 + \partial\Phi_3, \\ g_5 &= \sqrt{3}\partial\Phi_2 - \sqrt{3}\partial\Phi_3, \\ g_6 &= 2\partial\Phi_1 - \partial\Phi_2 - \partial\Phi_3. \end{aligned} \tag{4.4}$$

According to [27] and [43], Appendix 2, we demonstrate the motions generated by the Lie brackets, see Figures 8, 9 and 10. Let us stress that in spite of the simple car model example in Section 3.1 the trajectories do not represent the integral curves but are real trajectories of wheels and triangle vertices within the Euclidean plane. Further details of the Lie bracket exact realizations can be found in [27].

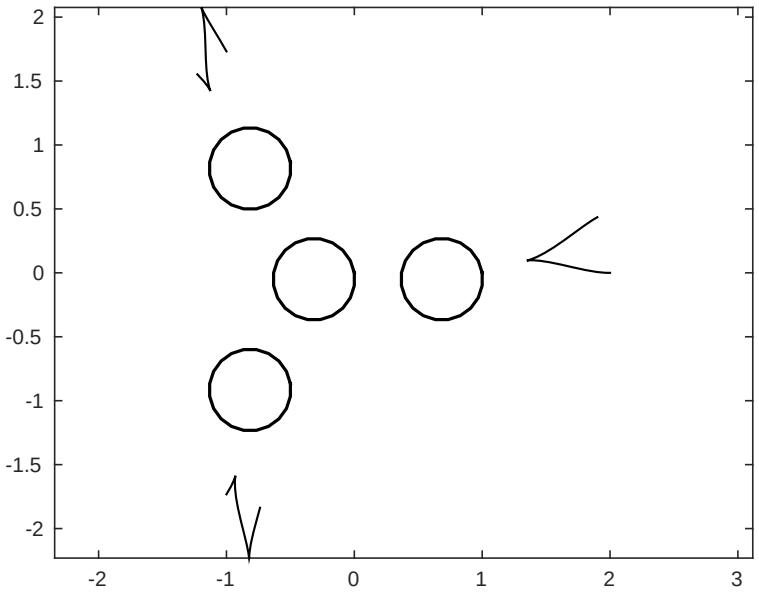


Figure 8: Realization of g_4 motion

Note that the trajectories on Figure 8 read that the root stays put and the angles represented by the coordinates Φ_1, Φ_2, Φ_3 change, which is obvious from approximately

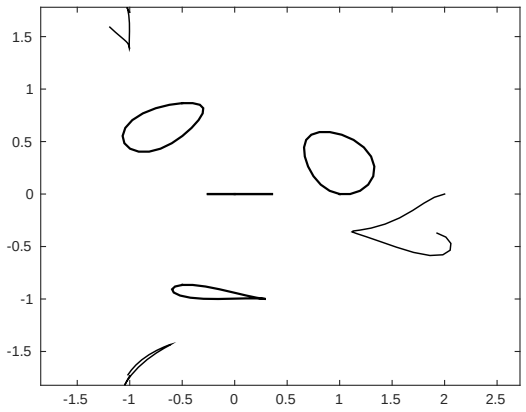


Figure 9: Realization of g_5 motion

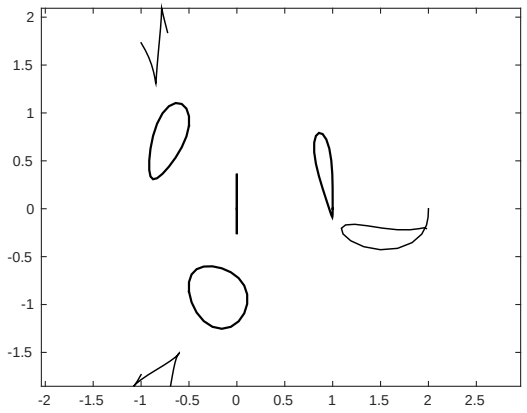


Figure 10: Realization of g_6 motion

equal dislocation of the wheel points at the end of the motion. Considering the vector field g_4 at 0 one finds that the angles should change proportionally to 1:1:1. Similarly, Figure 9 demonstrates the Lie bracket g_5 motion and clearly the trajectories represent the effect that the root moves along the x–axis and the angles change proportionally to 1:0:-1. Finally, Figure 10 shows g_6 realization which reads that the root moves along the y–axis and the angles change proportionally to -2:1:1.

This is crucial for the local motion planning algorithm as g_4 provides a shape transformation such that only the links move and the rest stays put and g_5 and g_6 provide motion along x and y axes, respectively. The local motion planning algorithm then runs in loops, i.e. once the spatial motion is performed and the position of the links is changed accordingly up to the construction limit, the bracket motion is realized to restore the original shape with no spatial effect and the mechanism is ready to repeat the motion in the desired direction. To find more details about the bracket motions and motion planning algorithm see [27, 28]. When compared to the simple car model, where the bracket motion corresponded to specific translation and rotation only and may be omitted for standard motion planning otherwise, it is clear that for snake like mechanisms the bracket motions are vital. Thus we come to a question of their realization which, according to [42], brings us to the periodic input and nilpotent approximation, see [43], Appendix 2. Indeed, the periodic input appropriate to the Lie bracket $[g_1, g_2]$ is a vector input function in the form

$$u(t) = (-A\omega \sin \omega t, A\omega \cos \omega t, 0) \tag{4.5}$$

with suitable choice of $A \in \mathbb{R}$ sufficiently small amplitude and $\omega \in \mathbb{R}$. For further details about the exact form of inputs for composed Lie brackets see [42].

4.2 3–link snake robot

Similarly to the previous section we describe the robot's kinematics. Note that we follow the notation used in [43, 22, 23, 24], Appendices 2–5.

The snake robot described in this text consists of 3 rigid links of constant length 2 interconnected by 2 motorized joints. To each line, in the centre of its mass, a pair of

wheels is attached which provides the same snake–like property as for the trident snake robot. To describe the actual position of a snake robot we need the set of 5 generalized coordinates

$$q = (x, y, \theta, \Phi_1, \Phi_2)$$

which describe the configuration of the snake robot as shown in Figure 11 and forms 5–dimensional manifold M as a phase space.

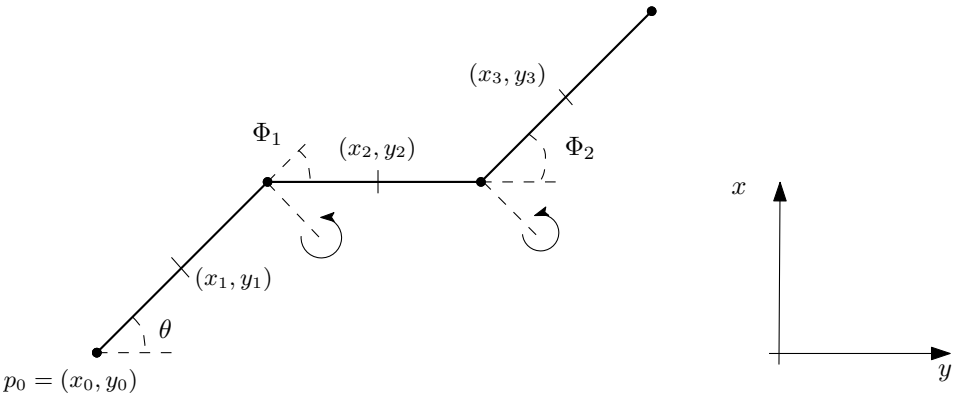


Figure 11: Snake robot model

Note that the forward kinematics is calculated w.r.t. the head point (x_0, y_0) and the model is parametrized by $\dot{x}$ and $\dot{y}$. The regular positions then form a 2–dimensional distribution which can be determined by the vector fields $X = (X_1, ..., X_6)$ and $Y = (Y_1, ..., Y_6)$ e.g. with the following coordinates:

$$\begin{aligned}
X_1 =& 1,\\
X_2 =& 0,\\
X_3 =& 2\sin(\theta),\\
X_4 =& -4\sin(\theta)\sin(\Phi_1+\pi/6) - 2\sin(\theta) - 2\cos(\theta+\Phi_1+\pi/6),\\
X_5 =& 8\cos(\Phi_2+\pi/3)\sin(\Phi_1+\pi/6)\sin(\theta) - 4\sin(\theta)\cos(\Phi_1+\Phi_2)\\
&+ 4\cos(\Phi_2+\pi/3)\cos(\theta+\Phi_1+\pi/6) + 4\sin(\theta)\sin(\Phi_1+\pi/6)\\
&+ 2\cos(\theta+\Phi_1+\pi/6) + 2\sin(\theta+\Phi_1+\Phi_2)
\end{aligned} \tag{4.6}$$

and

$$\begin{aligned}
Y_1 =& 0,\\
Y_2 =& 1,\\
Y_3 =& -2\cos(\theta),\\
Y_4 =& 4\cos(\theta)\sin(\Phi_1+\pi/6) + 2\cos(\theta) - 2\sin(\theta+\Phi_1+\pi/6),\\
Y_5 =& -8\cos(\Phi_2+\pi/3)\sin(\Phi_1+\pi/6)\cos(\theta) + 4\cos(\theta)\cos(\Phi_1+\Phi_2)\\
&+ 4\cos(\Phi_2+\pi/3)\sin(\theta+\Phi_1+\pi/6) - 4\cos(\theta)\sin(\Phi_1+\pi/6)\\
&+ 2\sin(\theta+\Phi_1+\pi/6) - 2\cos(\theta+\Phi_1+\Phi_2).
\end{aligned} \tag{4.7}$$

Their Lie bracket $[X,Y]$ and two second order brackets $[[X,Y],X]$ and $[[X,Y],Y]$ can be constructed, yet we omit their full coordinate form. Evaluating these vector fields at the point $init = (0,0,0,-\frac{\pi}{3},\frac{\pi}{3})$ which is considered in the following as the snake's initial position, we obtain

$$\begin{aligned} X(init) =& (1,0,0,-\sqrt{3},2\sqrt{3}),\\ Y(init) =& (0,1,-2,3,0),\\ [X,Y](init) =& (0,0,4,-12,12),\\ [[X,Y],X](init) =& (0,0,8,-36,60),\\ [[X,Y],Y](init) =& (0,0,0,-4\sqrt{3},20\sqrt{3}). \end{aligned}$$

The motion planning is then modelled sequentially, meaning that e.g. the motion in the direction of the x axis from the initial position and consequent transformation into the shifted initial position (i.e. only the x coordinate is different) again is formed by the linear combination

$$(1,0,0,0,0) \approx (576+360\sqrt{3})X - 22[X,Y] + 11[[X,Y],X] - 101\sqrt{3}[[X,Y],Y],$$

where the Lie bracket motions are realized by the periodic input u similar to (4.5) consisting of two components only, i.e. $u(t) = (-A\omega\sin\omega t, A\omega\cos\omega t)$. For further analysis and details see [43], Appendix 2.

5 Nilpotent Approximation

We construct the nilpotent approximation of a trident snake robot's controlling distribution. The aim is to show the construction that proves the periodic input (4.5) a convenient tool for modelling the Lie bracket motions, see [42]. Note that the construction of the 3–link snake robot's nilpotent approximation is described in [43], Appendix 2. We proceed according to Bellaïche's algorithm, see e.g. [29]. We stress that in [43], Appendix 2, one step of the algorithm is added which is valid for at least depth 3 filtrations. Recall that the trident snake's filtration is (3,6) and the 3–link robotic snake's filtration is (2,3,5). Thus we mainly point out the differences to [43], Appendix 2. In our particular case the configuration space of the trident snake robot is a 6–dimensional manifold M with the coordinate functions denoted by

$$(x,y,\theta,\Phi_1,\Phi_2,\Phi_3) =: (x_1,x_2,x_3,x_4,x_5,x_6).$$

Let the basis of a vector space T_pM be denoted by

$$(\partial_{x_1},\partial_{x_2},\partial_{x_3},\partial_{x_4},\partial_{x_5},\partial_{x_6}),\ p\in M$$

and let us consider three vector fields g_1,g_2,g_3 in the form (4.3) which determine a distribution in TM, and we add their Lie brackets g_4,g_5,g_6, see (4.4). For the introduction into the topic from the point of view of sub-Riemannian geometry see [29], for the notions needed within this text we refer to [43], Appendix 2. We recall the definition of privileged coordinates, [29].

Definition 5.1. A system of privileged coordinates at p is a system of local coordinates $(y_1, ..., y_n)$ such that $\mathrm{ord}_p(y_j) = w_j$ for $j = 1, ..., n$ where $\mathrm{ord}_p(y_j)$ stands for the order of the coordinate function y_j at a point p and w_j denotes the weights of the distribution.

The first question is what is the exact form of a coordinate transformation $x := (x_1, x_2, x_3, x_4, x_5, x_6) \rightsquigarrow (y_1, y_2, y_3, y_4, y_5, y_6) =: y$ such that the condition

$$\frac{\partial}{\partial y_i} \mid_p = g_i \mid_p, \quad i = 1, ..., 6 \tag{5.1}$$

holds in $p \in M$. Let us denote by $[g_k^i]_y$ the i–th coordinate of a vector g_k in the coordinate system y and by e^i a 6–dimensional vector with coordinates $e^i_j = 0$ for $i \neq j$ and $e^i_j = 1$ for $i = j, i, j \in \{1, ..., 6\}$. Then e.g. $[g_1^1]_x = 1$, $[g_1^2]_x = 0$, $[g_1^3]_x = 0$, $[g_1^4]_x = \sin(x_3 + x_6)$ etc. and the condition (5.1) reads $[g_i]_y = e^i$. Employing the Einstein summation convention, i.e. summing over j ranging from 1 to 6, the transformation law for vector fields under the coordinate change $x \rightsquigarrow y$ reads

$$\left[g_k^i\right]_y = \frac{\partial y_i}{\partial x_j} \left[g_k^j\right]_x .$$

Particularly, in the vector form we have

$$e^i = [g_i]_y = \begin{pmatrix} \frac{\partial y_1}{\partial x_j} \left[g_i^j\right]_x \\ \frac{\partial y_2}{\partial x_j} \left[g_i^j\right]_x \\ \vdots \\ \frac{\partial y_6}{\partial x_j} \left[g_i^j\right]_x \end{pmatrix} .$$

Evaluating all functions at an arbitrary point p, for sake of simplicity we choose the point $p = (0, 0, 0, 0, 0, 0)$, we get a system of 36 linear PDEs with respect to $\frac{\partial y_i}{\partial x_j}$ with constant coefficients. We split the system into 6 groups, each containing 6 equations for a particular y_i, determine the inverse matrix and continue by integration. Clearly, at an arbitrary $p \in M$ the desired transformation $x \rightsquigarrow y$ will be linear, in our case it will be given by

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & -2 & 1 & -1 & \sqrt{3} \\ 0 & -1 & -2 & 1 & 2 & 0 \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & -2 & 1 & -1 & -\sqrt{3} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} . \tag{5.2}$$

Proposition 5.2. *The coordinates $y = (y_1, y_2, y_3, y_4, y_5, y_6)$ form the system of privileged coordinates.*

For general proof see [29].

Vector fields g_i are of order ≥ -1 and thus generally their Taylor expansion is of the form:

$$g_i(y) \sim \sum_{\alpha,j} a_{\alpha,j} y^\alpha \partial_{y_j},$$

where $\alpha = (\alpha_1, ..., \alpha_n)$ is a multiindex. Furthermore, if we define a weighted degree of the monomial $y^\alpha = y_1^{\alpha_1} \cdots y_n^{\alpha_n}$ to be $w(\alpha) = w_1\alpha_1 + \cdots w_n\alpha_n$, then $w(\alpha) \geq w_j - 1$ if $a_{\alpha,j} \neq 0$. Recall that $w_j = \mathrm{ord}_p(y_j)$ from Definition 5.1 and in our particular case the coordinate weights are $(1, 1, 1, 2, 2, 2)$. Grouping together the monomial vector fields of the same weighted degree we express $g_i, i = 1, 2, 3$ as a series

$$g_i = g_i^{(-1)} + g_i^{(0)} + g_i^{(1)} + \cdots,$$

where $g_i^{(s)}$ is a homogeneous vector field of degree s. Note that this means that the $\partial_{y_1}, \partial_{y_2}$ and ∂_{y_3} coordinate functions of $g_1^{(-1)}, g_2^{(-1)}$ and $g_3^{(-1)}$ are formed by constants and the $\partial_{y_4}, \partial_{y_5}$ and ∂_{y_6} coordinate functions are linear polynomials in y_1, y_2, y_3. Then the following proposition holds, [29]:

Proposition 5.3. *Set $\hat{g}_i = g_i^{(-1)}, i = 1, 2, 3$. The family of vector fields $(\hat{g}_1, \hat{g}_2, \hat{g}_3)$ is a first order approximation of (g_1, g_2, g_3) at 0 and generates a nilpotent Lie algebra of step $r = 1$, i.e. all brackets of length greater than 1 are equal to zero.*

In our case, we obtain the following vector fields:

$$\begin{aligned}
\hat{g}_1 &= \partial_{y_1} - \frac{y_2}{2}\partial_{y_4} + (-\frac{y_2}{2} - y_3)\partial_{y_5} - \frac{y_1}{2}\partial_{y_6}, \\
\hat{g}_2 &= \partial_{y_2} + \frac{y_1}{2}\partial_{y_4} - \frac{y_1}{2}\partial_{y_5} + (\frac{y_2}{2} - y_3)\partial_{y_6}, \\
\hat{g}_3 &= \partial_{y_3}.
\end{aligned}$$

The family $(\hat{g}_1, \hat{g}_2, \hat{g}_3)$ is the nilpotent approximation of (g_1, g_2, g_3) at 0 associated with the coordinates y. The remaining three vector fields are generated by Lie brackets of $(\hat{g}_1, \hat{g}_2, \hat{g}_3)$ due to the second part of Proposition 5.3. Note that due to linearity of the three latter coordinates of $(\hat{g}_1, \hat{g}_2, \hat{g}_3)$, the coordinates of $(\hat{g}_4, \hat{g}_5, \hat{g}_6)$ must be constant. Indeed, we get $\hat{g}_4 = \partial_{y_4}$, $\hat{g}_5 = \partial_{y_5}$ and $\hat{g}_6 = \partial_{y_6}$. Note that the vector fields $(\hat{g}_1, \hat{g}_2, \hat{g}_3, \hat{g}_4, \hat{g}_5, \hat{g}_6)$ in $(x_1, x_2, x_3, x_4, x_5, x_6)$ coordinates are of the form

$$\begin{aligned}
\hat{g}_1 &= \partial_{x_1} - (x_2 + x_3)\partial_{x_4} - (\tfrac{\sqrt{3}x_1}{4} + \tfrac{x_2}{4} - \tfrac{x_3}{2} - \tfrac{\sqrt{3}}{2})\partial_{x_5} + (\tfrac{\sqrt{3}x_1}{2} - \tfrac{x_2}{4} + \tfrac{x_3}{2} - \tfrac{\sqrt{3}}{2})\partial_{x_6}, \\
\hat{g}_2 &= \partial_{x_2} - \partial_{x_4} + (\tfrac{3x_1}{4} + \tfrac{\sqrt{3}x_2}{4} - \tfrac{\sqrt{3}x_3}{2} + \tfrac{1}{2})\partial_{x_5} + (\tfrac{3x_1}{4} - \tfrac{\sqrt{3}x_2}{4} + \tfrac{\sqrt{3}x_3}{2} + \tfrac{1}{2})\partial_{x_6}, \\
\hat{g}_3 &= \partial_{x_3} - 2\partial_{x_4} - 2\partial_{x_5} - 2\partial_{x_6}, \\
\hat{g}_4 &= \partial_{x_4} + \partial_{x_5} + \partial_{x_6}, \\
\hat{g}_5 &= -\tfrac{\sqrt{3}}{2}\partial_{x_5} + \tfrac{\sqrt{3}}{2}\partial_{x_6}, \\
\hat{g}_6 &= -\partial_{x_3} + \tfrac{1}{2}\partial_{x_5} + \tfrac{1}{2}\partial_{x_6},
\end{aligned}$$

where the inverse transformation to (5.2) was applied. Once the nilpotent approximation is computed, it can be proved that the periodic input provides a model of the Lie bracket motions that is accurate up to order 2. We omit the proof within this text and refer to [42] for general proof.

6 Global motion planning algorithm

In 1990s, the term biorobotics left the area of science–fiction and became realistic mostly in medicine to replace the malfunctioning parts of a human body by an artificial substitute and, consequently, it turned out that the bio–inspired solutions are often the effective ones. Thus the applications shifted from mimicking the properties of the biological model into the development of nature inspired robotic mechanisms. We present a note about one possible global motion algorithm. This topic is well elaborated in [37] and is a part of [43], Appendix 2. A well–known bio–inspired mathematical description of lateral undulation was presented by Hirose in 1993 based on empirical studies of biological snakes. Hirose discovered that a close approximation to the shape of a biological snake during lateral undulation is given by a planar curve whose curvature varies sinusoidally, more precisely $\varkappa(s) = |ab\sin(bs) - c|$, see [37]. Hirose named it a serpenoid curve and described it by

$$\begin{aligned} x(s) &= \int_0^s \cos(a\cos(b\sigma) + c\sigma)\mathrm{d}\sigma, \\ y(s) &= \int_0^s \sin(a\cos(b\sigma) + c\sigma)\mathrm{d}\sigma \end{aligned}$$

where $(x(s), y(s))$ are the coordinates of the point along the curve at arc length s from the origin and a, b, c are positive scalars.

The pattern for lateral undulation is achieved by moving the joints of a planar snake robot according to

$$\Phi_i = A\sin(\omega t + (i-1)\delta) + \Phi_0,$$

where in our case, $i = 1, 2$, the offset $\Phi_0 = 0$ and only the amplitude A and frequency ω remain, see [37]. In [43], Appendix 2, we reparametrized the robotic system in such way that instead of controlling vector fields (4.6) and (4.7) parametrized by $\dot{x}$ and $\dot{y}$, we use vector fields parametrized by $\dot{\Phi}_1$ and $\dot{\Phi}_2$.

7 Conformal Geometric Algebra

7.1 Abstract structures in kinematics

The classical approach composes the kinematic chain of homogeneous matrices using the moving frame method and Euler angles, or, in a more advanced way, the quaternion algebra $\mathbb{H}$ which was introduced by W. R. Hamilton in 1843. By analogy with the complex numbers, quaternions are written as a real part and a linear combination of three complex units i, j, k as

$$q = a + b\cdot i + c\cdot j + d\cdot k$$

where $i^2 = j^2 = k^2 = ijk = -1$. Note that while quaternions are associative, they are not commutative. Indeed

$$ij = -ji = k,\ jk = -kj = i,\ ki = -ik = j.$$

Quaternions can be interpreted as a scalar plus a vector in $\mathbb{R}^3$ and this idea leads to the representation of an Euclidean point $x \in \mathbb{R}^3$ as an imaginary quaternion

$$x = (x_1, x_2, x_3) \rightleftharpoons x_1 i + x_2 j + x_3 k.$$

The crucial contribution of quaternions to mechanics is the representation of rotations in the Euclidean space. Indeed, if we consider $u = u_1 i + u_2 j + u_3 k$ such that $u\bar{u} = 1$ as the axis of rotation and q as a quaternion given by

$$q = \mathrm{e}^{\frac{\theta}{2}(u_1 i + u_2 j + u_3 k)} = \cos\tfrac{\theta}{2} + u \sin\tfrac{\theta}{2},$$

then the rotation of $x \in \mathbb{R}^3$ around the axis u is realized on the image of x in $\mathbb{H}$ by conjugation $x \mapsto qxq^{-1}$ provided that $q^{-1} = \mathrm{e}^{-\frac{\theta}{2}(u_1 i + u_2 j + u_3 k)}$, i.e. the problem of rotation in the three dimensional space reduces to the quaternion multiplication.

From the structural point of view the quaternions form a Clifford algebra $\mathcal{C}l(0,2)$ and thus it is only natural to search suitable generalizations among Clifford algebras. In 1984 D. Hestenes and G. Sobczyk in [14] described a suitable structure for object manipulation as Conformal Geometric Algebra referred to as CGA. Algebraically, we work with a geometric algebra on $\mathbb{R}^{4,1}$ which is the Clifford algebra $\mathcal{C}l(4,1)$ in which we can easily transform both linear objects and spheres of any dimension. Namely, these objects are simply elements of the algebra and can be transformed and intersected with ease. In addition, rotations, translation, dilations and sphere inversions all become rotations in our 2^5-dimensional space, see [18, 48, 61].

More precisely, let $\mathbb{R}^{4,1}$ denote a vector space $\mathbb{R}^5$ equipped with the scalar product of signature $(4,1)$ and let $\{e_1,\ e_2,\ e_3,\ e_+,\ e_-\}$ be an orthonormal basis. The Clifford algebra $\mathcal{C}l(4,1)$ can be described as a free, associative and distributive algebra such that the *geometric product* $e_i e_j$

(i) coincides with the scalar product in the case $i = j$,

(ii) is equal to $-e_j e_i$ for $i \neq j$. Hence the dimension of the algebra is $2^5 = 32$.

Next to the geometric product, we define two additional products on the elements of $\mathbb{R}^{4,1}$ based on the geometric one for any $u, v \in \mathbb{R}^{4,1}$, *inner product* and *wedge product*, respectively:

$$u \cdot v = \frac{1}{2}(uv + vu), \quad u \wedge v = \frac{1}{2}(uv - vu)$$

and thus for the base vectors we have $uv = u \cdot v + u \wedge v$. Note that $u \cdot v$ is a scalar (an algebra element of grade zero) while $u \wedge v$ is a bivector (an algebra element of grade two). In this sense, the definition of these product extends to the whole algebra. Namely, given two elements $E_k = e_{a_1} \wedge \cdots \wedge e_{a_k}$ and $E_l = e_{b_1} \wedge \cdots \wedge e_{b_l}$ of grades k and l, respectively, the wedge (outer) product is defined as

$$E_k \wedge E_l := \langle E_k E_l \rangle_{k+l}$$

while the inner product is defined as

$$E_k \cdot E_l := \langle E_k E_l \rangle_{|k-l|},$$

where $\langle \quad \rangle_k$ is the grade projection into the grade k. These products can be used effectively to compute an intersection of geometric objects and distances, respectively.

The conformal geometric basis elements can be represented by the multi–vectors from $\mathcal{C}l(4,1)$ either in the outer product null space (OPNS) representation or in its dual, so called inner product null space (IPNS) representation. To work with CGA effectively, one defines $e_0 = \frac{1}{2}(e_- - e_+)$ and $e_\infty = (e_- + e_+)$. Consequently, the following properties hold. The geometric objects which we use in the snake robot's kinematics description are then given as follows.

object	CGA element (OPNS)
Point	$Q = x + \frac{1}{2}x^2 e_\infty + e_0$
Point pair Q_1, Q_2	$P = Q_1 \wedge Q_2$
Line L	$L = Q_1 \wedge Q_2 \wedge e_\infty$
Circle C containing Q_1, Q_2, Q_3	$C = Q_1 \wedge Q_2 \wedge Q_3$

Each geometric transformation (rotation, translation, dilation, inversion) of a geometric object represented by an algebra element $\mathcal{O}$ is realized by conjugation $\mathcal{O} \mapsto M\mathcal{O}\tilde{M}$, where M is an appropriate multi–vector and $\tilde{M}$ is its reversion. For instance, the translation in the direction $t = t_1 e_1 + t_2 e_2 + t_3 e_3$ is realized by conjugation by the multi–vector

$$T = 1 - \tfrac{1}{2} t e_\infty,$$

which can be written as $e^{-\frac{1}{2}te_\infty}$, and the rotation around the axis L by angle φ is realized as conjugation by the multi–vector

$$R = \cos\tfrac{\varphi}{2} - L \sin\tfrac{\varphi}{2}.$$

Similarly to the case of a translation, the rotation can be also written as $\mathrm{e}^{-\frac{1}{2}\varphi L}$. The composition of translations and rotations is called the motor (the abbreviation of "momentum and vector," i.e. the element performing a screw motion) and usually denoted by M.

The direct kinematics for the snake robot is obtained similarly as the kinematics for serial robot arms [61]. In general, it is given by a succession of generalised rotations R_i and it is valid for all geometric objects, including point pairs. A point pair P in a general position is computed from its initial position P_0 as follows

$$P = \prod_{i=1}^{n} R_i P_0 \prod_{i=1}^{n} \tilde{R}_{n-i+1}.$$

Unlike the fixed serial robot arms, we allow R_i to be also a translation. We view translations as degenerate rotations.

7.2 Differential kinematics

By differential kinematics we understand the description of a motion not by the coordinates in the final state but by means of the velocity, i.e. by the norm and the direction of the velocity vector. As we consider both translations and rotations, both linear and angular velocities are involved. If we consider a robot whose final position is given by a kinematic chain (i.e. by the system of kinematic equations), the formulae for the differential kinematics are obtained simply as its total derivative. Let us consider the following kinematic chain in CGA:

$$M_n M_{n-1} \cdots M_1 P \tilde{M}_1 \cdots \tilde{M}_{n-1} \tilde{M}_n, \tag{7.1}$$

where M_i denotes the appropriate motors.

In the sequel, we consider the motors in the kinematic chain to be of the form

$$M_i = \mathrm{e}^{-\frac{1}{2} q_i L_i},$$

where L_i denotes the appropriate rotation axis and q_i are the angles of rotations, see [61]. At this point, by differentiation of the kinematic chain (7.1) we obtain [61, 12]

$$\dot{P} = \sum_{j=1}^{n} [P \cdot L'_j] dq_j, \tag{7.2}$$

where

$$L'_j = \prod_{i=1}^{j-1} M_i L_j \prod_{i=1}^{j-1} \tilde{M}_{j-i}.$$

In case of a planar robot model, e.g. a snake robot moving on a planar surface [23], the motors do not affect the axis L_j within the formula and thus we omit the function L'_j and leave only L_j instead. Let us remark that if we replace L_j with L'_j the following calculations are still valid and thus the results are true in full generality. Note that (7.2) represents the point P motion. In the rest of the paper we shall derive similar equations for other geometric objects, see [21] for proofs.

Lemma 7.1. *Let P_1 and P_2 be two moving points and let their final position be determined by the same kinematic chain* (7.1)*. Then the differential kinematics of the point pair $P_p = P_1 \wedge P_2$ is given by*

$$\dot{P}_p = \sum_{j=1}^{n} [P_p \cdot L'_j] dq_j.$$

Theorem 7.2. *Let P_1, P_2, P_3 and P_4 be four moving points whose final position is obtained by means of the same kinematic chain* (7.1)*. Then the differential kinematics of a 1D–sphere $S_1 = P_1 \wedge P_2 \wedge P_3$ and a 2D–sphere $S_2 = P_1 \wedge P_2 \wedge P_3 \wedge P_4$ are given by:*

$$\dot{S}_1 = \sum_{j=1}^{n} [S_1 \cdot L'_j] dq_j,$$

$$\dot{S}_2 = \sum_{j=1}^{n} [S_2 \cdot L'_j] dq_j.$$

Theorem 7.3. *Let P be a moving CGA geometric object, particularly a point, line, plane, point pair or a sphere of dimension 1 or 2, respectively. If the final position of P is determined by the kinematic chain* (7.1)*, then the differential kinematics of P is given by the equations* (7.2)*.*

The proof is a direct consequence of Lemma 7.1 and Theorem 7.2. The key fact is that a line is just a 1D–sphere S_1, where $P_3 = e_\infty$ and a plane is a 2D–sphere S_2, where $P_4 = e_\infty$.

The last object whose differential kinematics equations are useful for a robotic snake motion description is a sphere centre.

Lemma 7.4. *Let c be a centre of a sphere S (including a point pair as a 0D–sphere) whose final position is given by the kinematic chain* (7.1)*. Then the differential kinematics of c is given by*

$$\dot{c} = \sum_{j=1}^{n} [c \cdot L'_j] dq_j.$$

7.3 CGA and snake robot

Similarly to the beginning of the snake robot's construction, their description by CGA was inspired by robotic manipulators, see [12]. Each link is represented by the 0–dimensional sphere, referred to as a point pair, and the kinematic equations and control of the non–holonomic robotic snake is derived by means of CGA operations consequently. More precisely, to any link of a snake a single point pair is assigned and the mechanism is transformed by rotations and translations. Note that the coordinate and object notation is changed to comply with CGA description and with [22], Appendix 3.

We consider the snake robot model generally composed of n links of the lengths $l_i,\ i = 1, .., n$, interconnected by $(n-1)$ motorised joints with the appropriate axis of rotation denoted by $L_i,\ i = 1, .., n-1$, and each link is endowed with a pair of passive wheels at arbitrary position within the appropriate link $Q_i,\ i = 1, .., n-1$, see Figure 12. To describe such a complex model, it is suitable to use the tools of Conformal Geometric Algebra (CGA), where all the model modifications can be carried out quite easily, see [23], Appendix 4.

Note that a fixed coordinate system (x, y) is attached. For sake of simplicity, we consider the links to be of constant length 1 but the generalization to arbitrary lengths is obvious. The points $p_i := (x_i, y_i), i = 0, ..., n$, denote the endpoints of each link and by $Q_i = r_i p_i + (1 - r_i) p_{i-1},\ r_i \in \langle 0, 1 \rangle,\ i = 1, ..., n$, we denote the points where the wheels are attached to the particular link. Then, the distance $|Q_i p_{i-1}| = -2(Q_i \cdot p_{i-1})$ is equal to r_i. If the absolute angle of the i–th link, i.e. the angle between the link and the x–axis, is denoted by θ_i then the position of Q_i w.r.t. the global $x - y$ axes is then

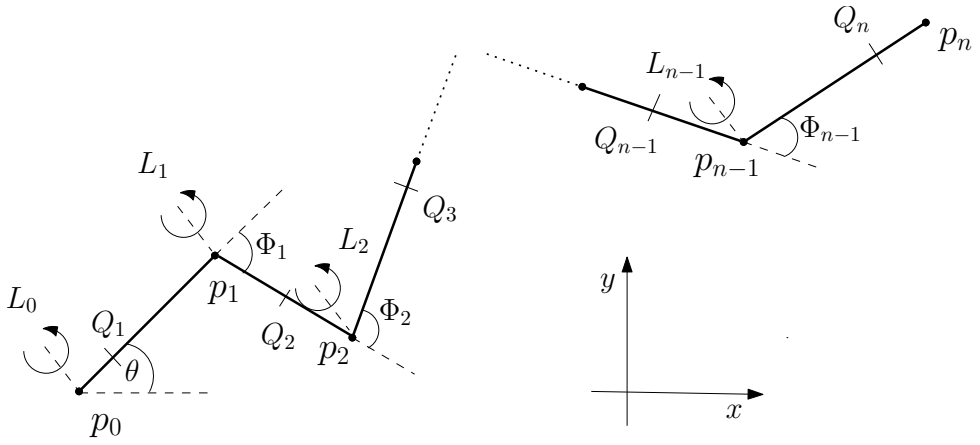


Figure 12: Snake robot model

expressed as

$$\begin{aligned} Q_{x,i} &= p_{x,0} + \sum_{j=1}^{i-1} \cos\theta_j + r_i \cos\theta_i, \\ Q_{y,i} &= p_{y,0} + \sum_{j=1}^{i-1} \sin\theta_j + r_i \sin\theta_i \end{aligned} \tag{7.3}$$

where by $p_{x,0}$ and $p_{y,0}$ we denote the x and y coordinate of the head point p_0, respectively. Note that to recover the generalized coordinates one has to consider the assertion

$$\theta_i = \sum_{j=1}^{i-1} \Phi_j + \theta.$$

Furthermore, the linear velocity of Q_i can be determined by taking the derivative of (7.3) and thus the nonholonomic equations are obtained.

$$\begin{aligned} \dot{Q}_{x,i} &= \dot{p}_{x,0} - \sum_{j=1}^{i-1} \sin\theta_j \dot{\theta}_j - r_i \sin\theta_i \dot{\theta}_i, \\ \dot{Q}_{y,i} &= \dot{p}_{y,0} + \sum_{j=1}^{i-1} \cos\theta_j \dot{\theta}_j + r_i \cos\theta_i \dot{\theta}_i. \end{aligned}$$

In [22], Appendix 3 we show the kinematics calculation for 3–linked robotic snake and in addition the dynamic terms are elaborated. Also the virtual model in CLUCalk software is provided. On the other hand, in [23], Appendix 4, we show the full contribution of CGA in describing completely general snake robot construction corresponding to Figure 12. We encourage the reader to compare this model to the 3–link snake robot from Section 4.2. We stress that CGA provides not only alternative analytical description of a mechanical system but may be used as a unifying tool for incorporating computer vision algorithms too, which is useful especially for autonomous robotic systems, see [61].

7.4 CGA and trident snake robot

Similarly to the previous section, by means of CGA quite general configuration of a trident snake robot may be described.

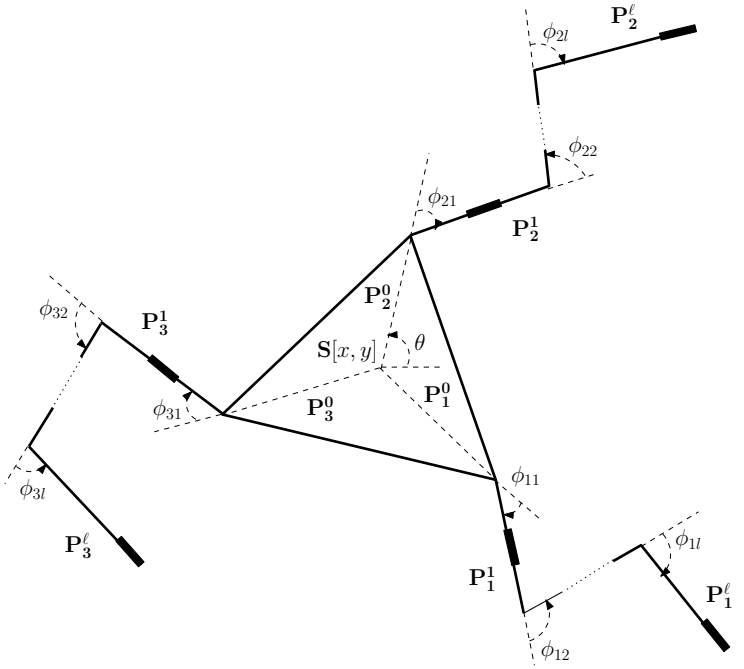


Figure 13: Trident snake robot model

Let us denote by Q_k^0 the centre S of the body, and by $Q_k^1, \cdots, Q_k^\ell$ the successive joints of the k–th branch leg, $k = 1, 2, 3$. As a central object that describes the state of the system we choose the set of point pairs which represent individual leg links

$$P = (P_1^0, \cdots, P_1^\ell, P_2^0, \cdots, P_2^\ell, P_3^0, \cdots, P_3^\ell),$$

see Figure 13. These point pairs are computed in terms of the wedge product in CGA as $P_k^i := Q_k^i \wedge Q_k^{i+1}$. At first, let us look at the zero position $q = 0$. Since $Q_2^i(0) = [i, 0]$, the elements in CGA corresponding to P_2^i are established as

$$\begin{aligned} P_2^i(0) &= \big(ie_1 + \tfrac{1}{2}i^2 e_\infty + e_0\big) \wedge \big((i+1)e_1 + \tfrac{1}{2}(i+1)^2 e_\infty + e_0\big) \\ &= \tfrac{1}{2}i(i+1)e_{1\infty} - e_{10} - \tfrac{1}{2}(2i+1)e_{\infty 0}, \end{aligned}$$

where we have used a shortened notation $e_{1\infty} = e_1 \wedge e_\infty$ etc. The algebra elements $P_{1,3}^i(0)$ which correspond to the zero position of the links of the first and the third branch leg are obtained from the corresponding links of the first branch by rotation by angle $\frac{2}{3}\pi$ and $-\frac{2}{3}\pi$ respectively, i.e.

$$P_{1,3}^i(0) = (\tfrac{1}{2} \pm \tfrac{\sqrt{3}}{2}e_{12})P_2^i(0)(\tfrac{1}{2} \mp \tfrac{\sqrt{3}}{2}e_{12}).$$

The particular point pairs in a general position q are obtained by a translation to $[x, y]$ composed by a trident body rotation θ and a series of rotations of the corresponding leg links by angles Φ_{ki}. For further details and consequent calculations of

the kinematic equation and differential kinematics of a trident snake robot in terms of CGA see [24], Appendix 5, where as a particular example, a 1–link trident snake robot is elaborated and a virtual model implemented in CLUCalc is also enclosed.

8 Further fields of interest

In this Section we recall several research areas and some author's results from these topics. Namely we mention the theory of jets and jet functors, higher order connections and some operations, especially the Ehresmann prolongation. To find the exact definitions of all the notions used within the following text such as the frame bundle, principal bundle, Lie group, fibred manifold etc. we refer to e.g. [32, 34].

8.1 Jets and jet functors

The theory of jets has been developed as a unifying tool for the description of many objects within differential geometry. For example, the frame bundle P^rM of a manifold M with dimension n can be viewed as a space $J^r_0(\mathbb{R}^n, M)$ of jets from $\mathbb{R}^n$ to M with source 0. Consequently, using the notion of jet prolongation of a fibred manifold, one can define even more complicated space, e.g. so called r-th order principal prolongation W^rP of a principal bundle $P \to M$ with structure Lie group G as $W^rP = P^rM \times J^rP$ with the appropriate structure Lie group W^r_mG, see [32, 34] for details. Note that principal prolongation of principal bundles plays a fundamental role in gauge theories of mathematical physics, see [10]. For geometric applications see [32].

To recall the definition of jets we first have to define the r–th order contact of two curves on a manifold M.

Definition 8.1. Two curves $\gamma, \delta : \mathbb{R} \to M$ have the r–th order contact at 0 if for every smooth function φ on M the difference $\varphi \circ \gamma - \varphi \circ \delta$ vanishes up to r–th order at $0 \in \mathbb{R}$.

Consequently, one can define the class of maps between two manifolds M and N determining the same jet.

Definition 8.2. Two maps $f, g : M \to N$ are said to determine the same r–jet at $x \in M$ if for every curve $\gamma : \mathbb{R} \to M$ with $\gamma(0) = x$ the curves $f \circ \gamma$ and $g \circ \gamma$ have the r–th order contact at zero.

An equivalence class of this relation is called an r–jet of M into N.

Now we can recall the definition of a jet prolongation of a fibred manifold. To define this notion, we first recall that a mapping $f : M \to N$ is called a submersion if its rank, i.e. the rank of the appropriate Jacobi matrix in local coordinates, at any point is equal to $\dim N$. Locally, the form of a submersion is that of a projection $\mathbb{R}^{n+k} \to \mathbb{R}^n$, in local coordinates $(x^1, \dots, x^n, x^{n+1}, \dots, x^{n+k}) \mapsto (x^1, \dots, x^n)$. In case of a tangent bundle TM we have a projection of two manifolds $p : TM \to M$.

Definition 8.3. A surjective submersion $p : Y \to M$ is called a fibred manifold with base M, total space Y and a projection p. The set $p^{-1}(x) =: Y_x \subset Y$ is called a fibre over the point $x \in M$. A smooth mapping $s : U \subseteq M \to Y$ is called a local section of a fibred manifold $p : Y \to M$ if $p \circ s = \mathrm{id}_M$.

Consequently, let $p : Y \to M$ be a fibred manifold. Denote by $J^rY \to M$ the r-th jet prolongation of $p : Y \to M$, that is the space of r-jets of local sections $M \to Y$. In what follows, J^rY will be called the r-th holonomic prolongation of Y. Recall that r-th nonholonomic prolongation $\widetilde{J}^rY$ of Y is defined by iteration

$$\widetilde{J}^1Y = J^1Y, \qquad \widetilde{J}^rY = J^1(\widetilde{J}^{r-1}Y \to M).$$

Clearly, we have an inclusion $J^rY \subset \widetilde{J}^rY$. Further, r-th semiholonomic prolongation $\overline{J}^rY \subset \widetilde{J}^rY$ is defined by the following induction. First, by $\beta_1 = \beta_Y$ we denote the projection $J^1Y \to Y$ and by $\beta_r = \beta_{\widetilde{J}^{r-1}Y}$ the projection $\widetilde{J}^rY = J^1\widetilde{J}^{r-1}Y \to \widetilde{J}^{r-1}Y$, $r = 2, 3, \ldots$. If we set $\overline{J}^1Y = J^1Y$ and assume we have $\overline{J}^{r-1}Y \subset \widetilde{J}^{r-1}Y$ such that the restriction of the projection $\beta_{r-1} : \widetilde{J}^{r-1}Y \to \widetilde{J}^{r-2}Y$ maps $\overline{J}^{r-1}Y$ into $\overline{J}^{r-2}Y$, we can construct $J^1\beta_{r-1} : J^1\overline{J}^{r-1}Y \to J^1\overline{J}^{r-2}Y$ and define

$$\overline{J}^rY = \{A \in J^1\overline{J}^{r-1}Y;\ \beta_r(A) \in \overline{J}^{r-1}Y\}.$$

Obviously, $J^r, \overline{J}^r$ and $\widetilde{J}^r$ are bundle functors on the category $\mathcal{FM}_{m,n}$ of fibred manifolds with m-dimensional bases and n-dimensional fibres and locally invertible fibre-preserving mappings.

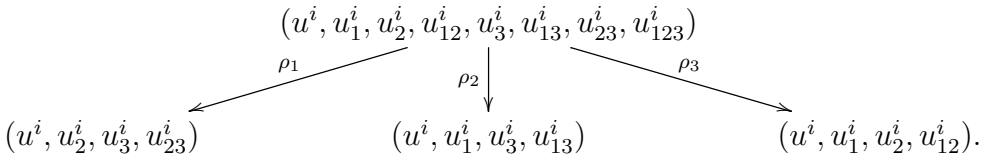
To find the author's results about existence and description of natural transformations between various jet functors, namely the transformations $\bar{J}^3 \to \bar{J}^3$, $\bar{J}^{\to}J^3$ and $\widetilde{J}^3 \to J^3$, we refer to [59].

8.2 Higher order connections

But not only the spaces can be viewed in the jet setting. We now mention the crucial object of differential geometry, i.e. that of a connection. Roughly speaking, a given connection defines the „direct lines“ in generally curved space. The motivation for this notion lies in parallel transport and geodesic curves on a manifold, see [32].

First, we recall rather geometric approach. A connection on a manifold M can be defined by means of the decomposition of the tangent bundle as follows, [52]. First, let us denote by T the tangent functor, by TM the tangent space of the manifold M and note that by T^kM we understand the k–th order tangent space defined iteratively as $T^kM = T(T^{k-1}M)$. A connection on bundle $\pi : Y \to M$ is defined by the structure $\triangle_h \oplus \triangle_v$ on a manifold Y where $\triangle_v = \ker T\pi$ is the vertical distribution tangent to the fibres and $\triangle_h$ is the horizontal distribution complementary to the distribution $\triangle_v$. The transport of the fibres along the path $\gamma \subset M$ is realized by the horizontal lifts given by the distribution $\triangle_h$ on the surface $\pi^{-1}(\gamma)$. If the bundle is a vector one and the transport of fibres along an arbitrary path is linear, then the connection is called linear.

A classical affine connection on manifold M is seen as a linear connection on the bundle $\pi_1 : TM \to M$. On the tangent bundle $TM \to M$ one can define the structure $\triangle_h \oplus \triangle_v$. To establish the notion of higher order connection in terms of horizontal distributions we recall that generally there exists k alternative projections $T^kM \to T^{k-1}M$ which will be denoted by $\rho_s,\ s = 1, ..., k$. For example, if the local coordinates on M are denoted by u^i, the induced coordinates on TM by (u^i, u^i_1) and on T^2M by $(u^i, u^i_1, u^i_2, u^i_{12})$ etc., the coordinate forms of three possible projections $\rho_s : T^3M \to T^2M$ are given by the following diagram:



Let us note that similarly to the jet prolongations one can consider the semiholonomic and holonomic case as well. We recall that the semiholonomity condition is connected to the notion of the osculating bundle, see [51], and can be defined as the equalizer of all possible projections.

Higher order connections are defined as follows: on tangent bundle TM the structure $\triangle \oplus \triangle_1$ is defined where $\ker T\rho_1 = \triangle_1$, on $T(TM)$ the structure $\Delta \oplus \Delta_1 \oplus \Delta_2 \oplus \Delta_{12}$ is defined where $\ker T\rho_s = \Delta_s \oplus \Delta_{12}\,, s = 1, 2$, etc. For further details see [52].

Once the notion of a distribution is taken into account, it presents a straightforward link to the dynamical systems and control theory as described in [53], Appendix 1.

On the other hand, one can define the connection on a fibred manifold as follows. By simple identification of horizontal subspaces of tangent space $T_yY,\ y \in Y$, with the elements $j^1_y s \in J^1_y Y,\ s : M \to Y$, a connection on the fibred manifold $Y \to M$ can be understood as a mapping $Y \to J^1Y$ to the first jet prolongation of Y. Passing to higher order jet prolongation of a fibred manifold one can define r-th order general connection Γ on $Y \to M$ as a mapping $\Gamma : Y \to J^rY$ to the r-th order jet prolongation of Y. There are three distinguished types of a connection appropriate to the considered target jet prolongations $\widetilde{J}^rY,\ \overline{J}^rY,\ J^rY$, i.e. nonholonomic, semiholonomic and holonomic, respectively.

Note that the author's results in the area of higher order connections can be found in [51, 52, 53]. In [58] we handle semiholonomic connections mainly as a result of so–called semiholonomic prolongation of a connection.

8.3 Ehresmann prolongation

The prolongation of a connection is motivated by quantum mechanics, higher order dynamics, field theory and gauge theories of mathematical physics. We mention a fundamental operator transforming a connection into higher order semiholonomic connection called Ehresmann prolongation. First, given two higher order connections $\Gamma : Y \to \widetilde{J}^rY$ and $\overline{\Gamma} : Y \to \widetilde{J}^sY$, the product of Γ and $\overline{\Gamma}$ is the $(r + s)$–th order

connection $\Gamma * \overline{\Gamma} : Y \to \widetilde{J}^{r+s}Y$ defined by

$$\Gamma * \overline{\Gamma} = \widetilde{J}^s\Gamma \circ \overline{\Gamma}.$$

If we consider two first order connections, the question of semiholonomity of their product was solved in [33, 34] in the following way: for two first order connections Γ and $\overline{\Gamma}$, the product $\Gamma * \overline{\Gamma} : Y \to \widetilde{J}^2Y$ is semiholonomic if and only if $\Gamma = \overline{\Gamma}$.

Considering a connection $\Gamma : Y \to J^1Y$, one can define an r–th order connection $\Gamma^{(r-1)} : Y \to \widetilde{J}^rY$ by

$$\Gamma^{(1)} := \Gamma * \Gamma = J^1\Gamma \circ \Gamma, \qquad \Gamma^{(r-1)} := \Gamma^{(r-2)} * \Gamma = J^1\Gamma^{(r-2)} \circ \Gamma.$$

The connection $\Gamma^{(r-1)}$ is called the $(r-1)$–st prolongation of Γ in the sense of Ehresmann, shortly $(r-1)$–st Ehresmann prolongation. Consequently, in [57] all natural operators transforming first order connection $\Gamma : Y \to J^1Y$ into second order semiholonomic connection $Y \to \overline{J}^2Y$ were described as a one parameter family

$$\Gamma \mapsto k \cdot (\Gamma * \Gamma) + (1-k) \cdot e(\Gamma * \Gamma), \qquad k \in \mathbb{R},$$

where $e : \overline{J}^2Y \to \overline{J}^2Y$ is obtained from the natural exchange map

$$e_\Lambda : J^1J^1Y \to J^1J^1Y$$

as a restriction to the subbundle $\overline{J}^2Y \subset J^1J^1Y$. As a next step, the properties of semiholonomic prolongations of higher order semiholonomic connections were considered.

The author's results in this field can be found in [57, 58]. Consequently, we applied the semiholonomity in the material science, see [25, 60].

References

[1] Atanasiu, Gh., Balan, V., Brînzei, N., Rahula, M.: *Differential-Geometric Structures. Tangent Bundles, Connections in Fiber Bundles, Exponential Law and Jet Spaces,* Librokom, Moscow, (in Russian), 2009.

[2] Bayraktaroglu, Z.Y.: *Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot,* Mech. Mach. Theory 44, No. 3, 591–602, 2009.

[3] Bayro-Corrochano, E., Daniilidis, K., Sommer, G.: *Motor algebra for 3D kinematics: The case of the hand-eye calibration,* Journal of Mathematical Imaging and Vision 13, 79–100, 2000.

[4] Carmo, do M.: *Differential Geometry of Curves and Surfaces,* Prentice-Hall, 1976.

[5] Chen, L., Ma, S., Wang, Y., Li, B.; Duan, D., *Design and modelling of a snake robot in traveling wave locomotion,* Mech. Mach. Theory 42, No. 12, 1632–1642, 2007.

[6] Chernousko, F. L.: *Modelling of snake-like locomotion,* Applied Mathematics and Computation, Vol. 164, Issue 2, 415–434, 2005.

[7] Chirikjian, G. S., Burdick, J. W.: *An obstacle avoidance algorithm for hyper-redundant manipulators,* In IEEE International Conference on Robotics and Automation, Cincinnati, 625–631, 1990.

[8] Crespi, A., Ijspeert, A. J.: *AmphiBot II: An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator,* In Proc. of 9th International Conference on Climbing and Walking Robots (CLAWAR 2006), 19–27, 2006.

[9] Dowling, K. J. : *Limbless Locomotion: Learning to Crawl with a Snake Robot,* PhD thesis, Carnegie Melon University, Pittsburgh, USA, 1997.

[10] Fatibene, M., Francaviglia, M.: *Natural and Gauge Natural Formalism for Classical Field Theories,* Kluwer, 2003.

[11] Fukuda, T., Hasegawa, Y., Sekiyama, K., Aoyama, T.: *Multi-locomotion robotic systems. New concepts of bio-inspired robotics,* Springer Berlin, 2012.

[12] Gonzalez–Jimenez, L., Carbajal–Espinosa, O., Loukianov, A., Bayro–Corrochano, E.: *Robust Pose Control of Robot Manipulators Using Conformal Geometric Algebra,* Advances in Applied Clifford Algebras, Vol. 24, No. 2, 533–552, 2014.

[13] Guo, X., Ma, SG., Li, B., Wang, MH., Wang, YC.: *Modeling and optimal torque control of a snake-like robot based on the fiber bundle theory,* Science China - Information Sciences, Vol. 58, Issue 3, 1–13, 2015.

[14] Hestenes, D., Sobczyk, G.: *Clifford Algebra to Geometric Calculus,* D. Reidel, Dordrecht/Boston, 1984.

[15] Hestenes, D., Li, H., Rockwood, A.: *Geometric Computing with Clifford Algebra,* Springer-Verlag Heidelberg, 2000.

[16] Hestenes, D., Li, H., Rockwood, A.: *New algebraic tools for classical geometry,* In Sommer, G. (ed.), Geometric Computing with Clifford Algebras, Vol. 40, 3-–23, 2001.

[17] Hestenes, D.: *Invariant body kinematics: I. Saccadic and compensatory eye movements,* Neural Networks 7, 65–77, 1994.

[18] Hildenbrand, D.: *Foundations of Geometric Algebra Computing,* Springer, Geometry and Computing, Vol. 8, 2013.

[19] Hirose, S., Umetani, Y.: *Kinematic control of active cord mechanism with tactile sensors,* In 2nd International CISM-IFTMM Symposium on Theory and Practice of Robots and Manipulators, 241–252, 1976.

[20] Hirose, S.: *Biologically inspired robots (snake-like locomotor and manipulator),* Oxford Uni. Press, 1993.

[21] Hrdina, J., Vašík, P.: *Notes on differential kinematics in conformal geometric algebra approach,* Mendel 2015, Advances in Intelligent Systems and Computing 378, 363–374, 2015.

[22] Hrdina, J., Návrat, A., Vašík, P.: *Control of 3–link robotic snake based on Conformal Geometric Algebra,* Adv. Appl. Clifford Algebras, Vol. 26, No. 3, 1069–1080, 2016.

[23] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: *CGA-based robotic snake control,* Adv. Appl. Clifford Algebras (in print), 1–12 online, 2016.

[24] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: *Geometric control of the trident snake robot based on CGA,* Adv. Appl. Clifford Algebras (in print), 1–13 online, 2016.

[25] Hrdina, J., Vašík, P.: *Semiholonomic second order connections associated to material bodies,* Journal of Applied Mathematics 2013 (1), 1–5, 2013.

[26] Ijspeert, A.J.: *Central pattern generators for locomotion control in animals and robots: A review,* Neural Networks, Vol. 21, 642–653, 2008.

[27] Ishikawa, M.: *Trident snake robot: Locomotion analysis and control,* In Proceedings of the IFAC NOLCOS, 1169–1174, 2004.

[28] Ishikawa, M., Minami, Y., Sugie, T.: *Development and control experiment of the trident snake robot,* IEEE/ASME Trans. on Mechatronics 15, 9–16, 2010.

[29] Jean, F.: *Control of Nonholonomic Systems: From Sub–Riemannian Geometry to Motion Planning,* Springer Briefs in Mathematics, Springer, 2014.

[30] Kane, T., Lecison, D.: *Locomotion of snakes: A mechanical 'explanation',* Int. J. Solids Struct., Vol. 37, No. 41, 5829—5837, 2000.

[31] Klaassen, B., Paap, K.L.: *GMD-SNAKE2: A snake-like robot driven by wheels and a method for motion control,* in ICRA 1999: Proc. of IEEE Int. Conf. on Robotics and Automation, 3014–3019, 1999.

[32] Kolář, I., Michor P. W., Slovák J.: *Natural Operations in Differential Geometry,* Springer-Verlag, 1993.

[33] Kolář, I.: *On the torsion of spaces with connections*, Czechoslovak Math. J. 21, 124–136, 1971.

[34] Kolář, I., Virsik, G.: *Connections in first principal prolongations,* Suppl. Rendiconti Circolo Mat. Palermo, Serie II, Vol. 43, 163–171, 1996.

[35] La Valle, S.M.: *Planning Algorithms,* Cambridge University Press, 2006.

[36] Lee, T., Ohm, T., Hayati, S.: *A highly redundant robot system for inspection,* In Proc. CIRFFSS 94, Houston, Texas, 142–149, 1994.

[37] Liljebäck, P., Pettersen, K.Y., Stavdahl, Ø., Gravdahl, J.T.: *Snake Robots, Modelling, Mechatronics and Control*, Springer, Advances in Industrial Control, 2013.

[38] Mahdavi, S. H., Bentley, P. J.: *Evolving motion of robots with muscles,* Applications of evolutionary computing, EvoWorkshops 2003, Springer, Lect. Notes Comput. Sci. 2611, 651–660, 2003.

[39] Miller, G.: *Chap. Snake Robots for Search and Rescue,* In Neurotechnology for Biomimetic Robots, MIT Press Cambridge, MA, USA, 271–284, 2002.

[40] Miller, G.: *Dr. Miller's snake robots,* http://www.snakerobots.com, Accessed: 29 September 2007.

[41] Morishima, A., Hirose, S.: *Design and control of mobile robot with an articulated body*, International Journal of Robotics Research, 9(2), 99–114, 1990.

[42] Murray, R. M., Zexiang, L., Sastry, S. S.: *A Mathematical Introduction to Robotic Manipulation,* CRC Press, 1994.

[43] Návrat, A., Vašík, P.: *On geometric control models of a robotic snake,* to appear in Note di Matematica.

[44] Perko, L.: *Differential Equations and Dynamic Systems*, Springer, 1991.

[45] Ostrowski, J.: *The Mechanics of Control of Undulatory Robotic Locomotion,* PhD thesis, CIT, 1995.

[46] Ostrowski, J.: *Computing reduced equations for robotic systems with constraints and symmetries,* IEEE Trans. Robot. Automat., Vol. 15, No. 1, 1999.

[47] Paap, K.L., Dehlwisch, M., Klaassen, B.: *GMD-snake: a semi-autonomous snake-like robot,* Distributed Autonomous Robotic Systems 2, Springer-Verlag, 1996.

[48] Perwass, Ch.: *Geometric Algebra with Applications in Engineering,* Springer, Geometry and Computing, Vol. 4, 2009.

[49] Prochazka, A., Ellaway, P.H.: *Sensory systems in the control of movement,* Comprehensive Physiology, John Wiley & Sons, American Physiological Society, 2615–2627, 2012.

[50] Rahula, M.: *New Problems in Differential Geometry,* World Scientific Publishing Co. Pte. Ltd., 1993.

[51] Rahula, M., Vašík, P., Voicu, N.: *Tangent structures: sector-forms, jets and connections*, J. Phys.: Conf. Ser., Vol. 346, 2012.

[52] Rahula, M., Vašík, P.: *A note on jet and geometric approach to higher order connections*, In Algebra, Geometry and Mathematical Physics, Springer Proceedings in Mathematics and Statistics, 375–385, 2014.

[53] Rahula, M., Vašík, P.: *Connections in control strategy,* Proceedings of the Estonian Academy of Sciences, Vol. 63, No. 1, 26–32, 2014.

[54] Selig, J.M.: *Geometric Fundamentals of Robotics,* Springer, Monographs in Computer Science, 2004.

[55] Spivak, M.: *A Comprehensive Introduction to Differential Geometry,* Third Edition, Volume 1, Publish or Perish, 2005.

[56] Umetani, Y., Hirose, S.: *An actice chord mechanism with oblique swivel joints and its control,* PWN-Polish Scientific Publishers, Zaborow, 327–340, 1981.

[57] Vašík, P.: *On the Ehresmann Prolongation,* Annales Universitatis Mariae Curie Sklodowska , Vol. LXI, Sectio A, 145–153, 2007.

[58] Vašík, P.: *Transformations of semiholonomic 2 and 3-jets and semiholonomic prolongation of connections,* Proceedings of the Estonian Academy of Sciences, Vol. 59, No. 4, 375–380, 2010.

[59] Vašík, P.: *On third order semiholonomic prolongation of a connection*, Banach Center Publications, Vol. 93, 271–276, 2011.

[60] Vašík, P.: *Higher order connections on Lie groupoid: application to materials,* Miskolc Mathematical Notes, Vol. 14, No. 2, 705–711, 2013.

[61] Zamora–Esquivel, J., Bayro–Corrochano, E.: *Kinematics and differential kinematics of binocular robot heads,* Robotics and Automation, 2006, ICRA 2006.

Appendix 1

[53] Rahula, M., Vašík, P.: *Connections in control strategy,* Proceedings of the Estonian Academy of Sciences, Vol. 63, No. 1, 26–32, 2014.

Proceedings of the Estonian Academy of Sciences,
2014, **63**, 1, 26–32
doi: 10.3176/proc.2014.1.05
Available online at www.eap.ee/proceedings

CONTROL
SYSTEMS

Connections in control strategy

Maido Rahula[a*] and Petr Vašík[b]

[a] Institute of Mathematics, University of Tartu, J. Liivi 2, 50409 Tartu, Estonia
[b] Institute of Mathematics, Brno University of Technology, Faculty of Mechanical Engineering, Technická 2, 616 69 Brno, Czech Republic; vasik@fme.vutbr.cz

Received 15 August 2013, revised 20 September 2013, accepted 14 October 2013, available online 14 March 2014

Abstract. We present an infinitesimal interpretation of the control theory, particularly of the part concerning dynamic systems. We use the original concept of a bundle connection, which lies in the idea of fibre transportation along a path on the base manifold. The control of a process leads also to the transportation of fibres, and the control strategy, i.e. the choice of a suitable system control in order to optimize the process corresponds to the choice of a path on the base manifold. The triple of crucial terms of control, aim–control–strategy, translates in the terms of connections as fibre–connection–curve. Such a scheme is quite convincing, but it also works well in dynamic systems analysis.

Key words: control theory, connection.

1. INTRODUCTION

When controlling a system, we not only apply one control model but also try to find a more suitable control model among the possible ones, i.e. we search the control strategy. We distinguish the following stages: controlled process – control correction – strategy choice.

Let us describe the mathematical setting. Let X, Y, and Z be three vector fields and let us denote by

$$a_t = \exp tX, \quad b_\sigma = \exp \sigma Y, \quad c_\tau = \exp \tau Z$$

the appropriate flows. If we understand the flow as a motion, the vector field can be seen as stopping the motion at the precise moment (stop-scene). Shortly, a vector field is an infinitesimal representation of the flow.

The flow c_τ of the vector field Z represents the controlled process (it is also possible to replace it by a transport of an arbitrary tensor field along the flow c_τ). Furthermore, the flow b_σ acts on the vector field Z flow c_τ as follows; see [1,3]:

$$c_\tau \rightsquigarrow b_\sigma c_\tau b_\sigma^{-1}, \quad Z \rightsquigarrow Z_\sigma \,.$$

This corresponds to the change of control. If in addition the flow a_t acts on b_σ, we have a control strategy

$$b_\sigma \rightsquigarrow a_t b_\sigma a_t^{-1}, \quad Y \rightsquigarrow Y_t \,.$$

* Corresponding author, rahula@ut.ee

Concerning the strategy $\{X \rightsquigarrow \{Y \rightsquigarrow Z\}\}$, it is obtained as a composition of the action of b_σ on c_τ and the action of a_t on b_σ,

$$c_\tau \rightsquigarrow b_\sigma c_\tau b_\sigma^{-1} \rightsquigarrow (a_t b_\sigma a_t^{-1}) c_\tau (a_t b_\sigma a_t^{-1})^{-1}.$$

The vector field Y plays the role of the one controlled by the vector field X and the role of the controlling field over Z.

The main goal of the paper is to describe the transformation of the parameters when the process Z is changed according to the strategy appropriate to the vector field X under the action of the vector field Y.

2. VECTOR FIELDS

Let M be a smooth manifold. The derivatives of a function $f : M \to \mathbb{R}$ along the vector fields X, Y, and Z are defined by

$$Xf \doteq (f \circ a_t)'_{t=0}, \quad Yf \doteq (f \circ b_\sigma)'_{\sigma=0}, \quad Zf \doteq (f \circ c_\tau)'_{\tau=0}.$$

The vector field Y is transported along the flow of X, which can be understood as an infinitesimal interpretation of such transportation (stop-scene) – the bracket of vector fields, i.e. Lie derivative $\mathscr{L}_X Y = [X,Y]$.

Remark 1. One can obtain the bracket of two vector fields $[X,Y] = XY - YX$ by double differentiation of a function f along the vector field flow $a_t b_\sigma a_t^{-1}$ w.r.t. σ and then w.r.t. t:

$$f \circ (a_t b_\sigma a_t^{-1})^{-1} \overset{(.)'_{\sigma=0}}{\longrightarrow} -\big(Y(f \circ a_t)\big) \circ a_t^{-1} \overset{(.)'_{t=0}}{\longrightarrow} (XY - YX) f\,.$$

Next, the transport of an arbitrary smooth tensor field along a vector field flow is defined by the Lie–Maclaurin series. For example, the transport of a vector field Z along the flow b_σ is defined by

$$Z \rightsquigarrow Z_\sigma = Z + Z'\sigma + Z''\frac{\sigma^2}{2} + \ldots = \sum_{k=0}^{\infty} Z^{(k)} \frac{\sigma^{(k)}}{k!},$$

where the coefficients $Z^{(k)} = \mathscr{L}_Y Z^{(k-1)}$, $k = 1,2,\ldots$, are Lie derivatives of Z with respect to Y.

In our situation, the vector field X plays three roles:

1. The vector field X itself causes the process as a motion in its flow a_t.
2. The operator $\mathscr{L}_X$ transforms the control $Y \rightsquigarrow [X,Y]$.
3. The operator $\mathscr{L}_{\mathscr{L}_X}$ defines the control strategy – control of control $\mathscr{L}_Y \rightsquigarrow [\mathscr{L}_X, \mathscr{L}_Y] = \mathscr{L}_{[X,Y]}$. Note that the above equality can be obtained from the Jacobi identity:

$$\big[[X,Y],Z\big] + \big[[Y,Z],X\big] + \big[[Z,X],Y\big] = 0$$

or equivalently

$$\big[[X,Y],Z\big] = \big[X,[Y,Z]\big] - \big[Y,[X,Z]\big].$$

Note that the process can be influenced only by some outer process, not by it itself. Indeed, if we admit that the operator $\mathscr{L}_X$ acts on the control by the process X, we obtain:

$$\mathscr{L}_X X = [X,X] = 0.$$

We assign the following operators to the vector fields X, Y, and Z:

1. the operator Z implementing the process (motion in the flow c_τ);
2. the operator $\mathscr{L}_Y$ implementing the control of the process $Z \rightsquigarrow [Y,Z]$ (motion of the flow c_τ in the flow b_σ);
3. the operator $\mathscr{L}_{\mathscr{L}_X}$ defining the control strategy $\mathscr{L}_Y$ (motion in the flow a_t of a motion of the flow c_τ in the flow b_σ).

3. CONTROL AND CONNECTION

We will follow the notations appropriate to the theory of connections on fibred manifolds; see, e.g., [1,4].

Let us consider a vector bundle $\pi : M_1 \to M$ with n-dimensional base manifold M and r-dimensional fibres. The standard fibre is isomorphic to $\mathbb{R}^r$. On a neighbourhood $U \subset M_1$ we have local coordinates (u^i, u^α), where (u^i) denotes the base coordinates and (u^α) the fibre coordinates. Precisely, $u^i = \bar{u}^i \circ \pi$, where $\bar{u}^i$ denotes the local coordinates on the neighbourhood $\pi(U) \subset M$. The coordinates (u^α) are the coordinates of $\mathbb{R}^r$. Latin indices $i, j, \ldots$ range from 1 to n, Greek indices $\alpha, \beta, \ldots$ range from $n+1$ to $n+r$.

We define two vector fields:

$$Y = y^\alpha \partial_\alpha \quad \text{and} \quad Z = z^\alpha \partial_\alpha .$$

Here z^α are the functions depending on the fibre coordinates u^α only, while y^α are the functions of all coordinates (u^i, u^α). The flow $b_\sigma = \exp \sigma Y$ is defined on the neighbourhood U by a system of ODEs

$$\frac{du^\alpha}{d\sigma} = y^\alpha(u^i, u^\beta). \tag{1}$$

Indeed, now we can see the connection between dynamic systems, see [2], and the controlling parameters (u^i). As mentioned above, these parameters are lifted from the base $\pi(U) \subset M$ to the neighbourhood $U \subset M_1$, i.e. $u^i = \bar{u}^i \circ \pi$.

On every fibre, the vector field Y induces a family of trajectories – phase portrait. When the fibre is changed, the vector field Y changes too and so does the phase portrait, i.e. the control $\{Y \rightsquigarrow Z\}$. A question arises: how do the parameters (u^i) affect the controlling process?

Let us consider the coordinate map

$$\Phi : (u^i, u^\alpha) \rightsquigarrow (u^i, s, I^\kappa),\ k = n+2, \ldots, n+r,$$

where s is a canonical parameter, i.e. $\mathscr{L}_Y s = 0$, and I^κ is a system of $r-1$ independent invariants of the vector field Y. The coordinates (u^i, I^κ) form a complete system of local invariants of Y on the manifold M_1. Now we can define the submersion of the manifold M_1 onto the fibre $\mathbb{R}^r$,

$$\varphi : M_1 \to \mathbb{R}^r : (u^i, u^\alpha) \rightsquigarrow (s, I^\kappa).$$

A fibre of the submersion φ has the dimension n and forms the family of the integral surfaces which define a horizontal distribution $\triangle_h$. Thus on the fibration π, a zero torsion connection structure $\triangle_h \oplus \triangle_v$ is defined. Let us consider the *adapted basis*

$$(X_i\ X_\alpha) = \left(\frac{\partial}{\partial u^j}\ \frac{\partial}{\partial u^\beta}\right) \cdot \begin{pmatrix} \delta_i^j & 0 \\ \Gamma_i^\beta & \delta_\alpha^\beta \end{pmatrix}, \quad \begin{pmatrix} \omega^i \\ \omega^\alpha \end{pmatrix} = \begin{pmatrix} \delta_j^i & 0 \\ -\Gamma_j^\alpha & \delta_\beta^\alpha \end{pmatrix} \cdot \begin{pmatrix} du^j \\ du^\beta \end{pmatrix},$$

where the vector fields

$$X_i = \frac{\partial}{\partial u^i} + \Gamma_i^\alpha \frac{\partial}{\partial u^\alpha}$$

form a base of the distribution Δ_h and the forms

$$\omega^\alpha = du^\alpha - \Gamma_i^\alpha du^i$$

vanish on the distribution Δ_h. The number of parameters Γ_i^α equals nr and they define the distribution Δ_h uniquely. On the other hand, the parameters Γ_i^α are determined by setting the functions φ^α equal to a constant on the fibres of the submersion φ, more precisely by their differentials:

$$d\varphi^\alpha = \varphi_i^\alpha du^i + \varphi_\beta^\alpha du^\beta = \varphi_\beta^\alpha (du^\beta + \bar{\varphi}_\gamma^\beta \varphi_i^\gamma du^i) = \varphi_\beta^\alpha \omega^\beta \quad \Longrightarrow \quad \Gamma_i^\alpha = \bar{\varphi}_\gamma^\alpha \varphi_i^\gamma,$$

where the coefficients of $d\varphi^\alpha$ are the partial derivatives of φ^α. The matrix (φ_β^α) is the integrating matrix with respect to the forms ω^α and its inverse is $(\bar{\varphi}_\alpha^\beta)$.

Theorem 1. *The vector field Y is projected by the submersion $\varphi : M_1 \to \mathbb{R}^r$ onto the vector field $T\varphi Y$ on the standard fibre $\mathbb{R}^r$. In the coordinates (s,I), where s denotes the canonical parameter and I is a system of the base invariants, the vector field $T\varphi Y$ represents the operator $\partial_s \doteq \dfrac{\partial}{\partial s}$. The vector field Z is expressed uniquely in the basis (∂_s, ∂_I) and the process controlled by Z is, in the coordinate system (s,I), described by the functions $s\circ\varphi$ and $I\circ\varphi$. These functions depend on the parameters u^α and the controlling parameters u^i.*

Proof. A family of the fibres corresponding to the submersion φ is defined by the solution of the system of differential equations $(u^\alpha)_\sigma = \varphi^\alpha(\sigma, u^i, u^\beta)$; see system (1). Furthermore, an arbitrary section of the fibration π can be extended into the system of imprimitivity appropriate to the flow b_s, i.e. the family of the fibres corresponding to the submersion φ. The vector field Y is φ-projected on the fibre $\mathbb{R}^r$. An integrable distribution $\Delta_h = \mathrm{Ker} T\varphi$ in the fibration π defines a zero curvature connection and thus on the neighbourhood U the basis and the co-basis of the distribution Δ_h is defined as follows:

$$X_i = \partial_i + \Gamma_i^\alpha \partial_\alpha\,, \quad \omega^\alpha = du^\alpha - \Gamma_i^\alpha du^i.$$

Let us recall that an arbitrary vector field $\bar{X}$ on the base manifold M can be lifted from M to the horizontal distribution Δ_h:

$$\bar{X} = \bar{x}^i \bar{\partial}_i \quad \rightsquigarrow \quad X = x^i X_i, \text{ where } x^i = \bar{x}^i \circ \varphi.$$

In our notations, the basis X_i represents the operators $\bar{\partial}_i$ from the neighbourhood $\pi(U)$ lifted to the distribution Δ_h.

It is now clear that the vector field X behaves with respect to the vector field Y as an infinitesimal symmetry, i.e. $[X,Y]=0$, and thus the impact on the vector field Y vanishes. In other words, the process appropriate to the vector field Z is defined on the fibre in the coordinates (s,I), where the functions $s\circ\varphi$ and $I\circ\varphi$ depend on the parameters u^α and the controlling parameters u^i. The vector field X affects the vector field Z indirectly by means of the invariants of the vector field Y. □

Remark 2. The components y^α of the vector field Y depend linearly and homogeneously on the fibre coordinates. Thus the defining system is described by the system of linear differential equations

$$\frac{du^\alpha}{d\sigma} = y^\alpha_\beta(u^i)u^\beta.$$

4. APPLICATION

On the bundle[1]

$$\pi : \mathbb{R}^3 \to \mathbb{R} : (u,x,y) \rightsquigarrow (u)$$

with the fibre coordinates (x,y) and the controlling parameter (or base coordinate) (u) we have the vector field

$$Y = \frac{\partial}{\partial x} + ux\frac{\partial}{\partial y}\,.$$

We define its flow $b_s = \exp sY$, the canonical parameter s, and the invariant I of Y as follows:

$$\begin{cases} \dot{x} = 1 \\ \dot{y} = ux \end{cases} \Rightarrow \begin{cases} x_s = x + s \\ y_s = y + u(xs + \frac{s^2}{2}), \end{cases} \qquad \begin{cases} s = x \\ I = y - \frac{ux^2}{2}\,. \end{cases}$$

We check that $\mathscr{L}_Y s = 1, \mathscr{L}_Y I = 0$. The trajectories on the fibres are parabolas depending on the parameter u.

[1] Here, for the sake of simplicity, we denote the local coordinates by (u,x,y) instead of (u^1,u^2,u^3) but note that the fibre coordinates (x,y) are in no way related to the components (x^i,y^α) of the vector fields X and Y.

The submersion $\varphi : \mathbb{R}^3 \to \mathbb{R} : (u,x,y) \rightsquigarrow (s,I)$ projects the space $\mathbb{R}^3$ onto the plane sI. The tangent mapping $T\varphi$ is defined by the following differentials and by the Jacobi matrix:

$$\begin{cases} ds = dx \\ dI = -\frac{x^2}{2}du - uxdx + dy, \end{cases} \qquad \begin{pmatrix} 0 & 1 & 0 \\ -\frac{x^2}{2} & -ux & 1 \end{pmatrix}.$$

The vector field Y with the components $(0,1,ux)$ is projected to the plane sI in which it forms the operator $T\varphi Y = \partial_s$ (see Fig. 1).

Thus on the bundle π a horizontal distribution

$$\triangle_h = \operatorname{Ker} T\varphi$$

is defined. The co-basis on $\triangle_h$ is of the form

$$\begin{cases} \omega^2 = & ds & = dx & = (dx - \Gamma_1^2 du) \\ \omega^3 = & uxds + dI = dy - \frac{x^2}{2}du & & = (dy - \Gamma_1^3 du), \end{cases}$$

and the connection coefficients are

$$\begin{pmatrix} \Gamma_1^2 \\ \Gamma_1^3 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{x^2}{2} \end{pmatrix}.$$

The adapted basis of the distribution $\triangle_h$ is characterized by the following:

$$X_1 = \partial_u + \frac{x^2}{2}\partial_y, \qquad \begin{pmatrix} \omega^2 \\ \omega^3 \end{pmatrix} = \begin{pmatrix} dx \\ dy \end{pmatrix} - \begin{pmatrix} 0 \\ \frac{x^2}{2} \end{pmatrix} \cdot (du).$$

The operator X_1 commutes with the vector field Y, i.e. $[X_1, Y] = 0$, and vanishes under the projection $T\varphi$, i.e. $T\varphi X_1 = 0$. The co-basis admits an integrating matrix as follows:

$$\begin{pmatrix} \omega^2 \\ \omega^3 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ us & 1 \end{pmatrix} \cdot \begin{pmatrix} ds \\ dI \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 \\ -ux & 1 \end{pmatrix} \cdot \begin{pmatrix} \omega^2 \\ \omega^3 \end{pmatrix} = \begin{pmatrix} ds \\ dI \end{pmatrix}.$$

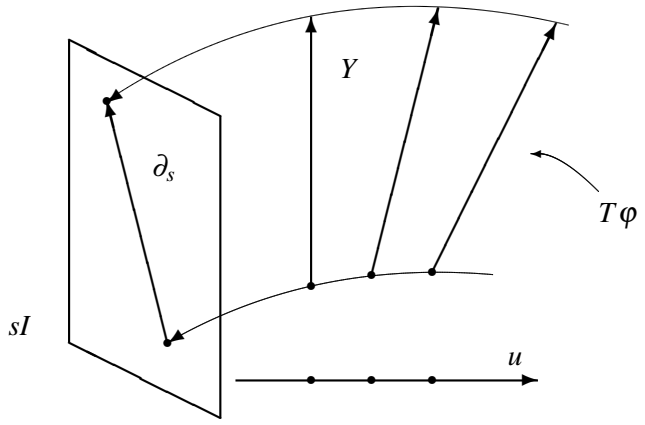


Fig. 1. Mapping $T\varphi : Y \to \partial_s$.

The direct impact of the parameter (u) on the operator Y is eliminated. Indeed, because the projection φ targets on the fibre xy, it is possible to change the coordinates under the condition $u = \text{const}$ from (x,y) to (s,I),

$$\begin{cases} x = s \\ y = \frac{us^2}{2} + I, \end{cases} \qquad \begin{pmatrix} 1 & 0 \\ us & 1 \end{pmatrix}.$$

Using the Jacobi matrix (the right-hand side), we can change the basis to the new natural one and we obtain the following frames and co-frames:

$$\left(\frac{\partial}{\partial x}\ \frac{\partial}{\partial y}\right) = \left(\frac{\partial}{\partial s}\ \frac{\partial}{\partial I}\right) \cdot \begin{pmatrix} 1 & 0 \\ -us & 1 \end{pmatrix}, \quad \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ us & 1 \end{pmatrix} \cdot \begin{pmatrix} ds \\ dI \end{pmatrix}.$$

Let us focus on the fibre. Note that the action of the vertical vector field Z can be understood as an action on a tensor field. Concerning the action of the operator Y on the vector field Z in the form

$$Y \rightsquigarrow Z = \mu \frac{\partial}{\partial x} + \nu \frac{\partial}{\partial y},$$

with the components (μ, ν), we can see that in new coordinates it reduces to the action of the operator ∂_s on the vector field $\tilde{Z}$ depending on the parameter u only:

$$\partial_s \rightsquigarrow \tilde{Z} = \mu \partial_s + \nu \partial_I - u\mu s\, \partial_I.$$

Note that Z and $\tilde{Z}$ are the same vector field, only expressed in the coordinates (x,y) and (s,I), respectively. The operators $T\varphi Y$ and ∂_s are the same operators expressed in different coordinate systems.

Thus we change the control:

$$\{Y \rightsquigarrow Z\} \quad \rightsquigarrow \quad \{\partial_s \rightsquigarrow \tilde{Z}\}.$$

Remark 3. As an example, let us consider the operator of rotation

$$Z = -y\frac{\partial}{\partial x} + x\frac{\partial}{\partial y}.$$

In coordinates (s,I), it can be written in the form $\tilde{Z} = -I\partial_s + s\partial_I + u(...)$, i.e. in such a form that some new operator with coefficient u is added. Such a property holds for an arbitrary linear dynamic system.

The control $\{Y \rightsquigarrow Z\}$ is described in the coordinates (x,y), while the control $\{\partial_s \rightsquigarrow \tilde{Z}\}$ is expressed in the coordinates (s,I). The parameter u affects the controlled field $\tilde{Z}$ directly.

5. CONCLUSION

The control of a dynamic system is viewed by means of differential geometry as the vector field Y on the bundle $\pi : M_1 \to M$ with the standard fibre $\mathbb{R}^r$ and the base manifold $M = \mathbb{R}^n$. The submersion φ is defined in such a way that the vector field Y is projected to the fibre $\mathbb{R}^r$. The distribution $\triangle_h = \text{Ker}T\varphi$ gives rise to the possibility of eliminating the dependence of the vector field Y on the controlling parameter u. The change of variables to (s,I), where s is the canonical parameter and I is the invariant of the field Y, changes the control $(Y \to Z)$ to the control $\{\partial_s \rightsquigarrow \tilde{Z}\}$, where the field ∂_s no longer depends on the parameter u while the controlled field $\tilde{Z}$ does so.

ACKNOWLEDGEMENTS

The first author was supported by Estonian Targeted Financing Project SF0180039s08. The second author was supported by the project NETME CENTRE PLUS (LO1202). The results of the project NETME CENTRE PLUS (LO1202) were co-funded by the Ministry of Education, Youth and Sports within the support programme "National Sustainability Programme I".

REFERENCES

1. Atanasiu, Gh., Balan, V., Brînzei, N., and Rahula, M. 2009. *Differential-Geometric Structures. Tangent Bundles, Connections in Fiber Bundles, Exponential Law and Jet Spaces*. Librokom, Moscow (in Russian).
2. Perko, L. *Differential Equations and Dynamic Systems*. Springer, 1991.
3. Rahula, M. *New Problems in Differential Geometry*. World Scientific Publishing Co. Pte. Ltd., 1993.
4. Rahula, M. and Vašík, P. A note on jet and geometric approach to higher order connections. *Springer Proc. Math. Statistics* (to appear).

Seostused juhtimise teoorias

Maido Rahula ja Petr Vašík

Juhtimisel on kolm aspekti: juhitav protsess, juhtiv protsess ja juhtimise valik/strateegia. Matemaatiliseks mudeliks on kihtkond, kus seostus mõjutab toimuvat kihil, ja baasiparameetrid, millest sõltub seostus. Need määravad juhtimise strateegia. Osutub, et baasiparameetreid võib seostuse abil otsekohe kihile suunata, st juhtivast protsessist juhitavasse.

Appendix 2

[43] Návrat, A., Vašík, P.: *On geometric control models of a robotic snake,* to appear in Note di Matematica.

Note di Matematica, manuscript, pages 1–10.

On geometric control models of a robotic snake

Aleš Návrat, Petr Vašík[i]
Brno University of Technology, Faculty of Mechanical Engineering
Institute of Mathematics
Technická 2896/2, 616 69 Brno
Czech Republic
navrat.a@fme.vutbr.cz, vasik@fme.vutbr.cz

Received: .....; accepted: .....

Abstract. We present three possible ways of controlling a robotic system called a 3–link snake robot. First, we recall the classical approach by constructing the controlling vector fields and their Lie brackets. Next we modify the coordinate system in order to obtain a nilpotent approximation of the controlling distribution. This is based on the notions of sub–Riemannian geometry. The third model is based on serpenoid curve and maintains the global control task.

Keywords: robotic snake, local control

MSC 2010 classification: 93B27

1 Introduction

Within this paper, we consider a 3–link snake robot moving on a planar surface. More precisely, it is a model when to each link a pair of wheels is attached and the joints of the legs are motorised and thus the possible motion directions are determined. Local controllability of such mechanism is known, see [4]. If the generalized coordinates are considered, the non–holonomic forward kinematic equations can be understood as a Pfaff system and thus a distribution on the configuration space is given. Rachevsky–Chow Theorem implies that the appropriate non–holonomic system is locally controllable if the corresponding distribution is not integrable and the span of the Lie algebra generated by the controlling distribution is of the same dimension as the configuration space. The spanned Lie algebra is then naturally endowed by a filtration which shows the way to realize the motions by means of the vector field brackets. In our case, the system is locally controllable and the appropriate filtration growth vector is $(2, 3, 5)$. The above considerations follow the geometric control ideas which is a

[i]This work was supported by a grant of the Czech Science Foundation (GAČR) no. 17–21360S.

modern geometrical approach to the control theory. For further applications of differential geometry in robotics see e.g. [6].

Note that to compose a motion control algorithm, the Lie bracket motions have to be considered to restore the initial position and allow the repetition of propulsion. As the Lie bracket motions are realized by means of so–called periodic input, an error occurs. To classify the model error, we establish so–called nilpotent approximation of the controlling distribution in which, according to [5], the periodic input models the Lie bracket motions that are accurate up to the second order. Furthermore, in e.g. [3], a convenient error estimates in the nilpotent approximation are described. Note that all constructions are local in the neighbourhood of 0 and the constructed nilpotent approximation is referred to as homogeneous.

2 Preliminaries

We recall the following concepts of functions or vector fields orders and distribution weights, see [3]. Let $X_1, ..., X_m$ denote the smooth vector fields on a manifold M and $C^\infty(p)$ denote the set of germs of smooth functions at $p \in M$. For $f \in C^\infty(p)$ we say that the Lie derivatives $X_i f, X_i X_j f, ...$ are non–holonomic derivatives of f of order 1,2,... The non–holonomic derivative of order 0 of f at p is $f(p)$.

Definition 1. Let $f \in C^\infty(p)$. Then the non–holonomic order of f at p, denoted by $\mathrm{ord}_p(f)$, is the biggest integer k such that all non–holonomic derivatives of f of order smaller than k vanish at p.

Note that in case $M = \mathbb{R}^n$, $m = n$ and $X_i = \partial_{x_i}$, for a smooth function f, $\mathrm{ord}_0(f)$ is the smallest degree of monomials having nonzero coefficient in the Taylor series. In the language of non–holonomic derivatives, the order of a smooth function is given by the formula, [3]:

$$\mathrm{ord}_p(f) = \min\left\{ s \in \mathbb{N} : \exists i_1, ..., i_s \in \{1, ..., m\} \text{ s.t. } (X_{i_1} \cdots X_{i_s} f)(p) \neq 0 \right\},$$

where the convention reads that $\min \emptyset = \infty$.

If we denote by $\mathrm{VF}(p)$ the set of germs of smooth vector fields at $p \in M$, the notion of non–holonomic order extends to the vector fields as follows:

Definition 2. Let $X \in \mathrm{VF}(p)$. The non–holonomic order of X at p, denoted by $\mathrm{ord}_p(X)$, is a real number defined by:

$$\mathrm{ord}_p(X) = \sup\left\{ \sigma \in \mathbb{R} : \mathrm{ord}_p(Xf) \geq \sigma + \mathrm{ord}_p(f), \forall f \in C^\infty(p) \right\}.$$

Note that $\mathrm{ord}_p(X) \in \mathbb{Z}$. Moreover, the null vector field $X \equiv 0$ has infinite order, $\mathrm{ord}_p(0) = \infty$. Furthermore, $X_1, ..., X_m$ are of order ≥ -1, $[X_i, X_j]$ of order ≥ -2, etc. Using the notion of a vector field order one can define

Definition 3. A family of m vector fields $(\hat{X}_1, ..., \hat{X}_m)$ defined near p is called a first order approximation of $(X_1, ..., X_m)$ at p if the vector fields $X_i - \hat{X}_i, i = 1, ..., m$ are of order ≥ 0 at p.

Finally, to define the weights of distributions we use the same notation as in [3]. Let us by Δ^1 denote the distribution $\Delta^1 = \mathrm{span}\{X_1, ..., X_m\}$ and for $s \geq 1$ define $\Delta^{s+1} = \Delta^s + [\Delta^1, \Delta^s]$, where $[\Delta^1, \Delta^s] = \mathrm{span}\{[X, Y] : X \in \Delta^1, Y \in \Delta^s\}$. Then $\Delta^s = \mathrm{span}\{X_I : |I| \leq s\}$. Note that this directly leads to the fact that every $X \in \Delta^s$ is of order $\geq -s$. Now let us consider the sequence $\Delta^1(p) \subset \Delta^2(p) \subset \cdots \subset \Delta^{r-1} \subsetneq \Delta^r(p) = T_pM$, where $r = r(p)$ is called the degree of non–holonomy at p. Set $n_i(p) = \dim \Delta^i(p)$. Then we can define the weights at p, $w_i = w_i(p), i = 1, ..., n = n_{r(p)}$ by setting $w_j = s$ if $n_{s-1}(p) < j \leq n_s(p)$, where $n_0 = 0$. In other words, we have

$$w_1 = \cdots = w_{n_1} = 1,\ w_{n_1+1} = \cdots = w_{n_2} = 2, ..., w_{n_{r-1}+1} = \cdots = w_{n_r} = r.$$

The weights at p form an increasing sequence $w_1(p) \leq \cdots \leq w_n(p)$.

3 3–link snake robot

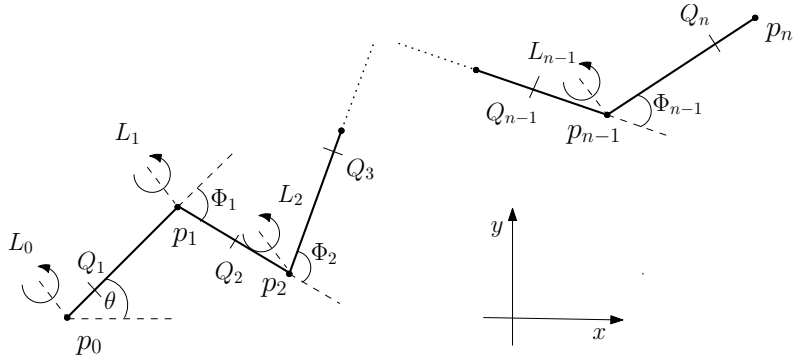


Figure 1. Snake robot model

The snake robot model is generally composed of n links of the lengths $l_i,\ i = 1, .., n$, interconnected by $(n-1)$ motorised joints with the axis of rotation denoted by $L_i,\ i = 1, .., n-1$, and each link is endowed with a pair of passive wheels at arbitrary position within the appropriate link $Q_i,\ i = 1, .., n-1$, see 1. To describe such a complex model, it is suitable to use the tools of Conformal

Geometric Algebra (CGA), where all the model modification can be carried on quite easily, see [1].

The snake robot described in this paper consists of 3 rigid links of constant length 2 interconnected by 2 motorized joints. To each line, in the centre of its mass, a pair of wheels is attached to provide an important snake-like property that the ground friction in the direction perpendicular to the link is considerably higher than the friction of a simple forward move. In particular, this prevents the slipping sideways. To describe the actual position of a snake robot we need the set of 5 generalized coordinates $q = (x, y, \theta, \Phi_1, \Phi_2)$ which describe the configuration of the snake robot as shown in Figure 2 and forms a manifold M as a phase space.

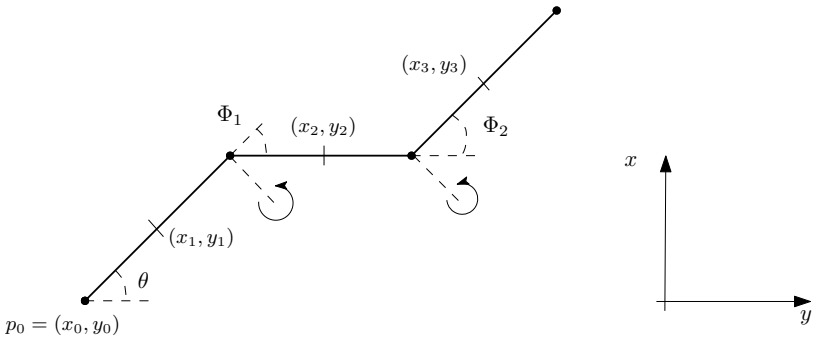


Figure 2. 3–link snake robot model

Note that the forward kinematics is calculated w.r.t. the head point (x_0, y_0) and the parameterization by $\dot{x}$ and $\dot{y}$. The non–singular positions then form a 2–dimensional distribution which can be determined by the vector fields X, Y e.g. with the following coordinates:

$$
\begin{aligned}
X_1 =& 1 \\
X_2 =& 0 \\
X_3 =& 2\sin(\theta) \\
X_4 =& -4\sin(\theta)\sin(\Phi_1 + \pi/6) - 2\sin(\theta) - 2\cos(\theta + \Phi_1 + \pi/6) \\
X_5 =& 8\cos(\Phi_2 + \pi/3)\sin(\Phi_1 + \pi/6)\sin(\theta) - 4\sin(\theta)\cos(\Phi_1 + \Phi_2) \\
& + 4\cos(\Phi_2 + \pi/3)\cos(\theta + \Phi_1 + \pi/6) + 4\sin(\theta)\sin(\Phi_1 + \pi/6) \\
& + 2\cos(\theta + \Phi_1 + \pi/6) + 2\sin(\theta + \Phi_1 + \Phi_2)
\end{aligned}
\tag{1}
$$

and

$$
\begin{aligned}
Y_1 =& 0 \\
Y_2 =& 1 \\
Y_3 =& -2\cos(\theta) \\
Y_4 =& 4\cos(\theta)\sin(\Phi_1+\pi/6)+2\cos(\theta)-2\sin(\theta+\Phi_1+\pi/6) \\
Y_5 =& -8\cos(\Phi_2+\pi/3)\sin(\Phi_1+\pi/6)\cos(\theta)+4\cos(\theta)\cos(\Phi_1+\Phi_2) \\
&+4\cos(\Phi_2+\pi/3)\sin(\theta+\Phi_1+\pi/6)-4\cos(\theta)\sin(\Phi_1+\pi/6) \\
&+2\sin(\theta+\Phi_1+\pi/6)-2\cos(\theta+\Phi_1+\Phi_2)
\end{aligned} \tag{2}
$$

And their Lie bracket $[X,Y]$ and two higher order brackets $[[X,Y],X]$ and $[[X,Y],Y]$. Evaluating these vector fields at the point $init = (0,0,0,-\frac{\pi}{3},\frac{\pi}{3})$ which is considered in the following as the snake's initial position, we obtain

$$
\begin{aligned}
X(init) =& (1,0,0,-\sqrt{3},2\sqrt{3}), \\
Y(init) =& (0,1,-2,3,0), \\
[X,Y](init) =& (0,0,4,-12,12), \\
[[X,Y],X](init) =& (0,0,8,-36,60), \\
[[X,Y],Y](init) =& (0,0,0,-4\sqrt{3},20\sqrt{3}).
\end{aligned}
$$

The motion planning is then modelled sequentially, meaning that e.g. the motion in the direction of the x axis from the initial position and consequent transformation into the shifted initial position (i.e. only the x coordinate is different) again is formed by the linear combination

$$(1,0,0,0,0) \propto (576+360\sqrt{3})X - 22[X,Y] + 11[[X,Y],X] - 101\sqrt{3}[[X,Y],Y],$$

where the Lie bracket motions are realized by so–called periodic input in the form

$$u(t) = (-A\omega\sin\omega t, A\omega\cos\omega t) \tag{3}$$

with suitable choice of $A \in \mathbb{R}$ sufficiently small amplitude and $\omega \in \mathbb{R}$. For further details about the precise form of inputs for composed Lie brackets see [5].

4 Nilpotent Approximation

We proceed according to Bellaïche's algorithm, see e.g. [3]. We shall use the following notation: $(x_1,x_2,x_3,x_4,x_5) := (x,y,\theta,\Phi_1,\Phi_2)$ and the resulting privileged coordinates will be denoted by (z_1,z_2,z_3,z_4,z_5). We recall the definition of privileged coordinates, [3], taking into account the notation from Section 2.

Definition 4. A system of privileged coordinates at p is a system of local coordinates $(y_1, ..., y_n)$ such that $\mathrm{ord}_p(y_j) = w_j$ for $j = 1, ..., n$.

Yet, note that the algorithm starts with the coordinate transformation into $(y_1, y_2, y_3, y_4, y_5)$ such that $\partial_{y_1} = X$, $\partial_{y_2} = Y$, $\partial_{y_3} = [X, Y]$, $\partial_{y_4} = [[X, Y], X]$ and $\partial_{y_5} = [[X, Y], Y]$. The transformation matrix at 0 is of the form

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1/4\sqrt{3} & 5/4 & 5/4 & 5/12 & 1/12 \\ -1/8\sqrt{3} & -3/8 & -1/2 & -5/24 & -1/24 \\ 1/8 & 1/8\sqrt{3} & 1/4\sqrt{3} & 1/8\sqrt{3} & 1/24\sqrt{3} \end{pmatrix}.$$

If the controlling vector fields are denoted as

$$\begin{aligned} g_1 &:= X, \\ g_2 &:= Y, \\ g_3 &:= [X, Y], \\ g_4 &:= [[X, Y], X], \\ g_5 &:= [[X, Y], Y], \end{aligned}$$

then generally the condition reads

$$\frac{\partial}{\partial y_i}|_p = g_i(p), \ i = 1, \ldots, 5$$

where $p \in M$. Note that set (y_i) is then called an adapted frame at p.

Note that we keep the notation of the transformed controlling vector fields to be g_i and, if needed, we use the notation $g_i(z)$ to denote the vector fields transformed into $(z_1, z_2, z_3, z_4, z_5)$ coordinate system etc.

Next step of the Bellaïche's algorithm is the following: For $j = 1, \ldots, 5$ set

$$z_j = y_j - \sum_{k=2}^{w_j - 1} h_k(y_1, \ldots, y_{j-1}), \tag{4}$$

where, for $k = 2, \ldots, w_j - 1$,

$$h_k(y_1, \ldots, y_{j-1}) = \sum_{\substack{|\alpha|=k \\ w(\alpha)<w_j}} g_1^{\alpha_1} \ldots g_{j-1}^{\alpha_{j-1}} \left(y_j - \sum_{q=2}^{k-1} h_q(y) \right) (p) \frac{y_1^{\alpha_1}}{\alpha_1!} \cdots \frac{y_{j-1}^{\alpha_{j-1}}}{\alpha_{j-1}!}.$$

Note that the choice of the polynomials h_k in (4) guarantees that the non–holonomic derivatives of z_j at p up to order $w_j - 1$ vanish. Clearly, only the

coordinates of the order 3, i.e. z_4 and z_5 will be different from the adapted coordinates y_4 and y_5, the lower order coordinates remain unchanged. Note that this step is employed due to the depth of the filtration (2,3,5) while for e.g. the trident snake robot, [2], with the filtration (3,6) this step is omitted. In our case,

$$z_4 = y_4 + 3y_1y_2 + (1/2)\sqrt{3}y_1^2 - (5/2)\sqrt{3}y_2^2,$$
$$z_5 = y_5 - 12y_1y_2 + \sqrt{3}y_1^2 + 5\sqrt{3}y_2^2.$$

Proposition 1. *The coordinates $(z_1, z_2, z_3, z_4, z_5)$ form the system of privileged coordinates.*

For general proof see [3].

To form the nilpotent approximation Vector fields g_i are of order ≥ -1 and thus generally their Taylor expansion is of the form:

$$g_i(z) \sim \sum_{\alpha,j} a_{\alpha,j} z^\alpha \partial_{z_j},$$

where $\alpha = (\alpha_1, ..., \alpha_n)$ is a multiindex. Furthermore, if we define a weighted degree of the monomial $z^\alpha = z_1^{\alpha_1} \cdots z_n^{\alpha_n}$ to be $w(\alpha) = w_1\alpha_1 + \cdots w_n\alpha_n$, then $w(\alpha) \geq w_j - 1$ if $a_{\alpha,j} \neq 0$. Furthermore, the weighted degree of the monomial vector field $z^\alpha \partial z_j$ is $w(\alpha) - w_j$. Recall that $w_j = \mathrm{ord}_p(z_j)$ from Definition 4 and in our particular case the coordinate weights are $(1, 1, 2, 3, 3)$. Grouping together the monomial vector fields of the same weighted degree we express $g_i, i = 1, 2$ as a series

$$g_i = g_i^{(-1)} + g_i^{(0)} + g_i^{(1)} + \cdots,$$

where $g_i^{(s)}$ is a homogeneous vector field of degree s. Note that this means that the controlling vector fields coefficients of ∂_{z_1} and ∂_{z_2} are formed by constants, the coefficients of ∂_{z_3} are linear in z_1 and z_2 and independent of the rest, and finally, the coefficients of ∂_{z_4} and ∂_{z_5} are formed by polynomials of the weighted degree 2, i.e. quadratic in ∂_{z_1} and ∂_{z_2} and linear in ∂_{z_3}. This fully corresponds to the fact that the weights of the coordinates are (1,1,2,3,3), see Section 2 for explanation. Then the following proposition holds, [3]:

Proposition 2. *Set $\hat{g}_i = g_i^{(-1)}, i = 1, 2$. The family of vector fields $(\hat{g}_1, \hat{g}_2)$ is a first order approximation of (g_1, g_2) at 0 and generates a nilpotent Lie algebra of step $r = 2$, i.e. all brackets of length greater than 2 are zero.*

The proof is just a straightforward computation of Lie brackets and is obvious. Because of its very extensive form we show the coordinate form of the

approximated vector field corresponding to X only and we denote it by $\hat{X}$. Note that it is expressed in the original coordinate system $(x, y, \theta, \Phi_1, \Phi_2)$.

$$
\begin{aligned}
\hat{X} = \\
&\partial_x \\
&- \frac{1}{2(3\sqrt{3}-2)^2}\Big(621\sqrt{3}x^2 - 13832\sqrt{3}xy - 12705\sqrt{3}y^2 + 120\sqrt{3}\Phi_1 + 24\sqrt{3}\Phi_2 \\
&\quad - 186\sqrt{3}x + 264\sqrt{3}y + 360\sqrt{3}\theta + 644x^2 + 21626xy + 21580y^2 \\
&\quad -310\Phi_1 - 62\Phi_2 + 216x - 682y - 930\theta\Big)\partial_\theta \\
&- \frac{1}{(3\sqrt{3}-2)^3}\Big(-1164\sqrt{3}x^2 - 149430\sqrt{3}xy - 145220\sqrt{3}y^2 + 1755\sqrt{3}\Phi_1 \\
&\quad + 351\sqrt{3}\Phi_2 - 1700\sqrt{3}x + 4212\sqrt{3}y + 5265\sqrt{3}\theta - 7044x^2 + 272028xy \\
&\quad +252600y^2 - 170\sqrt{3} - 2550\Phi_1 - 510\Phi_2 + 3510x - 6120y - 7650\theta + 351\Big)\partial_{\Phi_1} \\
&- \frac{1}{2(3\sqrt{3}-2)^3}\Big(1290\sqrt{3}x^2 + 106170\sqrt{3}xy + 99950\sqrt{3}y^2 - 680\sqrt{3}x + 5925x^2 \\
&\quad -204180xy - 163425y^2 + 680\sqrt{3} + 1404x - 1404\Big)\partial_{\Phi_2}
\end{aligned}
$$

The motion planning algorithm is similar to the one proposed in Section 3, i.e. it is sequential with periodic input (3) applied for the bracket motions. Note that according to [5], the periodic input (3) in nilpotent approximation can be applied to model the Lie bracket motions.

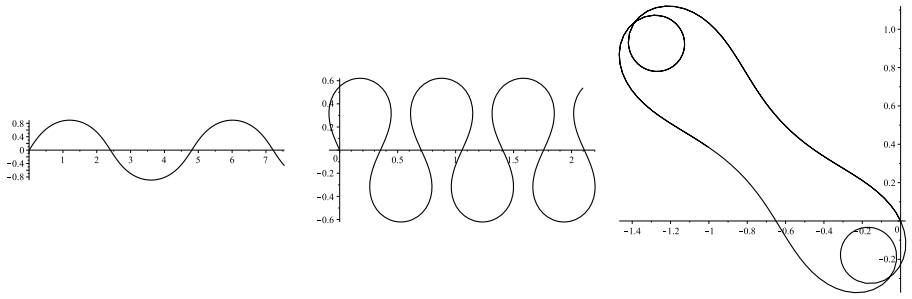
5 Global control

A well–known mathematical description of lateral undulation was presented by Hirose in 1993 based on empirical studies of biological snakes. Hirose discovered that a close approximation to the shape of a biological snake during lateral undulation is given by a planar curve whose curvature varies sinusoidally, more precisely $\varkappa(s) = |ab\sin(bs) - c|$, see [4]. Hirose named it a serpenoid curve and described it by

$$
\begin{aligned}
x(s) &= \textstyle\int_0^s \cos(a\cos(b\sigma) + c\sigma)\mathrm{d}\sigma, \\
y(s) &= \textstyle\int_0^s \sin(a\cos(b\sigma) + c\sigma)\mathrm{d}\sigma
\end{aligned}
$$

where $(x(s), y(s))$ are the coordinates of the point along the curve at arc length s from the origin and a, b, c are positive scalars. The following pictures within Figure 3 show the form of a serpenoid curve for different choices of parameters.

Furthermore, it was shown by Sainto in 2002 that a serpenoid curve of arc length 1 can be approximated by N identical discrete segments by calculating the angle Φ_i



(a) a=1,b=1,c=0 (b) a=2,b=2,c=0 (c) a=2,b=2,c=3

Figure 3. Serpenoid curves

of segment $i \in \{1, ..., N\}$ with respect to the x axis according to $\Phi_i = a\cos\left(\frac{ib}{N}\right)$. This implies that a snake robot with N identical discrete links attains a discrete approximation to the serpenoid curve by moving its ling angles sinusoidally with a constant phase shift between the links. The pattern for lateral undulation is achieved by moving the joints of a planar snake robot according to

$$\Phi_i = A\sin(\omega t + (i-1)\delta) + \Phi_0, \tag{5}$$

where in our case, $i = 1, 2$, the offset $\Phi_0 = 0$ and only the amplitude A and frequency ω remain, see [4]. To use this model, we have to reparametrize the robotic system in such way that instead of controlling vector fields (1) and (2) parametrized by $\dot{x}$ and $\dot{y}$, we use vector field parametrized by $\dot{\Phi}_1$ and $\dot{\Phi}_2$. Then we use (5) as an input. Figure 4 shows the trajectory of a head point within one second.

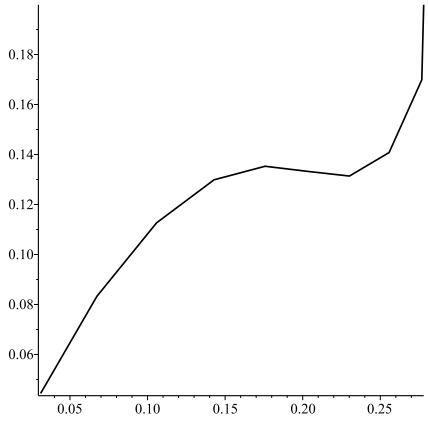


Figure 4. Head point trajectory

Finally, to demonstrate the similarity of the nilpotent approximation with the original controlling model see Figure 5. Note that due to locality property of the nilpotent

approximation the time for comparison was decreased to 0,25 second.

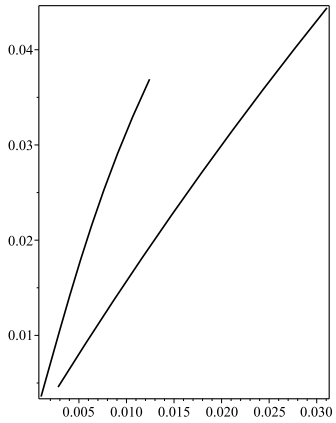


Figure 5. Nilpotent approximation comparison

6 Conclusion

We presented three possible models of a 3–link snake robot control, two of them local and one global. Note that if the motion is realized on a smooth surface with no obstacles then the control the robot is realized by the global control model. If the obstacles and narrow places are added such that the number of actually operational motorized joints decreases under the critical value of 4 then the local control takes place. We conclude that the questions of optimality are yet to be solved.

References

[1] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: *CGA-based robotic snake control*, Adv.Appl. Clifford Algebr. (in print), 1–12 online (2016)

[2] Hrdina, J., Návrat, A., Vašík, P.: *Nilpotent approximation of a trident snake robot controlling distribution*, arXiv:1607.08500.

[3] Jean, F.: *Control of Nonholonomic Systems: From Sub–Riemannian Geometry to Motion Planning*, Springer Briefs in Mathematics, Springer, 2014.

[4] P. Liljebäck, K.Y. Pettersen, Ø. Stavdahl, J.T. Gravdahl, *Snake Robots, Modelling, Mechatronics and Control*, Springer, Advances in Industrial Control, 2013.

[5] Murray,R. M., Zexiang,L., Sastry, S. S.: *A Mathematical Introduction to Robotic Manipulation*, CRC Press, 1994.

[6] Selig, J.M.: *Geometric Fundamentals of Robotics*, Springer, Monographs in Computer Science, 2004.

Appendix 3

[22] Hrdina, J., Návrat, A., Vašík, P.: *Control of 3–link robotic snake based on Conformal Geometric Algebra,* Adv.Appl. Clifford Algebras, Vol. 26, No. 3, 1069–1080, 2016.

Adv. Appl. Clifford Algebras 26 (2016), 1069–1080

0188-7009/031069-12
published online November 12, 2015
DOI 10.1007/s00006-015-0621-2

Advances in Applied Clifford Algebras

Control of 3-Link Robotic Snake Based on Conformal Geometric Algebra

Jaroslav Hrdina*, Aleš Návrat and Petr Vašík

Abstract. Local controllability of a three link robotic snake is solved by means of 5D conformal geometric algebra. The non-holonomic kinematic equations are assembled, their role in the geometric control theory is discussed and the control solution is found. The functionality is demonstrated on a virtual model in CLUCalc programme. Finally, the snake robot dynamics is elaborated.

Mathematics Subject Classification. Primary 93C10; Secondary 15A66.

Keywords. Conformal geometric algebra, Clifford algebra, Mathematics robotic, Nonholonomic mechanics, Snake robots, Local controllability, Bionics.

1. Introduction

Within this paper, we consider a three-link snake robot moving on a planar surface. More precisely, it is a model when to each link a pair of wheels is attached and thus the possible movement directions are determined uniquely. The aim is to find the complete local controllability solution. If generalized coordinates are considered, the non-holonomic forward kinematic equations can be understood as a Pfaff system and its solution as a distribution in the configuration space. Rachevsky–Chow Theorem implies that the appropriate non-holonomic system is locally controllable if the corresponding distribution is not integrable and the span of the Lie algebra generated by the controlling distribution has to be of the same dimension as the configuration space. The spanned Lie algebra is then naturally endowed by a filtration which shows the way to realize the movements by means of the vector field brackets [6,8]. In our case, the system is locally controllable and the filtration is $(2, 3, 5)$.

The classical approach composes the kinematic chain of homogeneous matrices using the moving frame method and Euler angles [4]. Instead of this, our aim is to use the notions of conformal geometric algebra, i.e. the subset of a Clifford algebra $\mathcal{C}l(4, 1)$ where the Euclidean space $\mathbb{E}_3$ is included by a

The authors were supported by a Grant No. FSI-S-14-2290.
*Corresponding author.

mapping $x \mapsto x + x^2 e_\infty + e_0$. In this geometric setting, we can easily handle both linear objects and spheres of dimensions 2, 1 and 0, see [2,7,9].

In particular, the 0-dimensional sphere, referred to as a point pair, is used to derive the kinematic equations and for the control of the robotic snake, consequently. More precisely, to any link of a snake a single point pair is assigned and the mechanism is transformed by rotations and translations. We introduce the differential kinematic equations (3), as well as the non-holonomic conditions (5) or (7), respectively. Also the singularity condition was formulated. Furthermore, the system movement dynamic equations are derived (14). We demonstrate the functionality in the CLUCalc software designed for the computations in Clifford algebra, particularly in conformal geometric algebra.

2. Conformal Geometric Algebra—CGA

Let $\mathbb{R}^{4,1}$ denote a vector space $\mathbb{R}^5$ equipped with the scalar product of signature $(4,1)$ and thus we have the corresponding Clifford algebra $\mathcal{C}l(4,1)$ such that the set $\{e_1, e_2, e_3, e_+, e_-\}$ is the basis. To describe the elements of $\mathcal{O} := \mathcal{C}l(4,1)$ we have to determine a free, associative and distributive algebra as a span of the set $\{e_1, e_2, e_3, e_+, e_-\}$ such that the following identities are satisfied:

$$e_1^2 = e_2^2 = e_3^2 = e_+^2 = 1, \;\; e_-^2 = -1,$$

$$e_i e_j = -e_j e_i, \; i \neq j, \; i,j \in \{1,2,3,+,-\}.$$

In this case, we get $2^5 = 32$-dimensional vector space.

TABLE 1. Elements of $\mathcal{C}l(4,1)$

Scalars	1
Vectors	e_1, e_2, e_3, e_+, e_-
Bivectors	$e_1e_2, e_1e_3, e_1e_+, e_1e_-, e_2e_3, e_2e_+, e_2e_-, e_3e_+, e_3e_-, e_-e_+$
Threevectors	$e_1e_2e_3, e_1e_2e_+, e_1e_2e_-, e_1e_3e_+, e_1e_3e_-, e_1e_+e_-, e_2e_3e_+,$ $e_2e_3e_-, e_2e_+e_-, e_3e_+e_-$
Fourvectors	$e_1e_2e_3e_+, e_1e_2e_3e_-, e_1e_2e_+e_-, e_1e_3e_+e_-, e_2e_3e_+e_-$
Pseudoscalars	$e_1e_2e_3e_+e_-$

Let us note that the *geometric product* in $\mathbb{R}^{4,1}$ coincides with the scalar product and the norm in $\mathbb{R}^{4,1}$ can be understood as a vector square $x^2 = \|x\|^2$.

Now, we define two additional products on $\mathbb{R}^{4,1}$ based on the geometric one for any $u, v \in \mathcal{O}$, *dot product* and *wedge product*, respectively:

$$u \cdot v = \frac{1}{2}(uv + vu), \quad u \wedge v = \frac{1}{2}(uv - vu)$$

and thus the geometric product of the basis elements are derived as

$$uv = u \cdot v + u \wedge v.$$

Generally, the wedge (outer) product of two basis blades E_i and E_j, with $k = \mathrm{gr}(E_i)$, $l = \mathrm{gr}(E_j)$ is defined as

$$E_i \wedge E_j := \langle E_i E_j \rangle_{k+l}$$

and the dot (inner) product is defined as

$$\begin{aligned} E_i \cdot E_j &:= \langle E_i E_j \rangle_{|k-l|}, \;\; i, j, > 0 \\ &:= 0, \;\; i = 0 \text{ or } j = 0, \end{aligned}$$

where $\mathrm{gr}(E)$ is a grade of the basis blade E and $\langle \;\; \rangle_k$ is the grade projection into grade k. To work with CGA effectively, we have to define a new basis of $\mathbb{R}^{4,1}$ as a set $\{e_1, e_2, e_3, e_0, e_\infty\}$ such that $e_0 = \frac{1}{2}(e_- + e_+)$ and $e_\infty = (e_- - e_+)$. Consequently, the following properties hold:

$$e_0^2 = 0, \quad e_\infty^2 = 0, \quad e_\infty e_0 = -1 - e_- \wedge e_+, \quad e_0 e_\infty = -1 + e_- \wedge e_+,$$
$$e_\infty e_0 = -e_0 e_\infty - 2.$$

In CGA, we can represent the points by the following multivector from $\mathcal{C}l(4,1)$:

$$\text{point } x \rightsquigarrow Q = x + \frac{1}{2}x^2 e_\infty + e_0$$

Note that the previous object is expressed in the IPNS representation while for other objects such as spheres and point pairs in particular we use the OPNS representation that is more convenient. Indeed, their expression in OPNS is the following:

$$\begin{aligned} \text{point pair } Q_1, Q_2 &\rightsquigarrow P = Q_1 \wedge Q_2. \\ \begin{matrix}\text{circle containing} \\ \text{the points } Q_1, Q_2, Q_3\end{matrix} &\rightsquigarrow C = Q_1 \wedge Q_2 \wedge Q_3. \end{aligned}$$

In CGA (in fact in GA generally), any transformation of the element O is realized by conjugation

$$O \mapsto T O \tilde{T}$$

where T is the appropriate multivector from $\mathcal{O}$. For instance, the translation in the direction $t = t_1 e_1 + t_2 e_2 + t_3 e_3$ is realized by the multivector

$$T = 1 - \frac{1}{2} t e_\infty$$

and the rotation around the axis L by angle ϕ is realized by the multivector

$$R = \cos\frac{\phi}{2} - L \sin\frac{\phi}{2}$$

where $L = a_1 e_2 e_3 + a_2 e_1 e_3 + a_3 e_1 e_2$.

3. Control Theory

The snake robot described in this paper consists of 3 rigid links of constant length 2 interconnected by 2 motorized joints. To each line, in the center of mass, a pair of wheels is attached to provide an important snake-like property that the ground friction in the direction perpendicular to the link is considerably higher than the friction of a simple forward move. In particular, this prevents the slipping sideways. To describe the actual position of a snake robot we need the set of 5 generalized coordinates

$$q = (x, y, \theta, \Phi_1, \Phi_2) \tag{1}$$

which describe the configuration of the snake robot as shown in Fig. 1.

Note that a fixed coordinate system (x, y) is attached. The points $p_1 := (x_1, y_1)$, $p_2 := (x_2, y_2)$, $p_3 := (x_3, y_3)$ denote the centers of mass of each link. To describe the robotic snake we use as a central object the couple of point pairs

$$(P_1, P_3)$$

where $P_1 = Q_1 \wedge Q_2$ and $P_3 = Q_3 \wedge Q_4$, where Q_i are the joints. Consequently, the kinematic equations can be assessed and if we consider the projections

$$Q_2 = -\frac{\sqrt{P_1 \cdot P_1} + P_1}{e_\infty \cdot P_1}, \quad Q_3 = \frac{\sqrt{P_3 \cdot P_3} + P_3}{e_\infty \cdot P_3},$$

we are able to express the first point coordinates from any point pair. Finally, denote $P_2 = Q_2 \wedge Q_3$. The coordinates of a particular position vectors are expressed as

$$\begin{aligned} p_1 &= P_1 e_\infty \tilde{P}_1, \quad \text{s.t.} \quad P_1 = R_\theta T_{x,y} P_{1,0} \tilde{T}_{x,y} \tilde{R}_\theta, \\ p_2 &= P_2 e_\infty \tilde{P}_2, \quad \text{s.t.} \quad P_2 = R_{\Phi_1} R_\theta T_{x,y} P_{2,0} \tilde{T}_{x,y} \tilde{R}_\theta \tilde{R}_{\Phi_1}, \\ p_3 &= P_3 e_\infty \tilde{P}_3, \quad \text{s.t.} \quad P_3 = R_{\Phi_2} R_{\Phi_1} R_\theta T_{x,y} P_{3,0} \tilde{T}_{x,y} \tilde{R}_\theta \tilde{R}_{\Phi_1} \tilde{R}_{\Phi_2} \end{aligned} \tag{2}$$

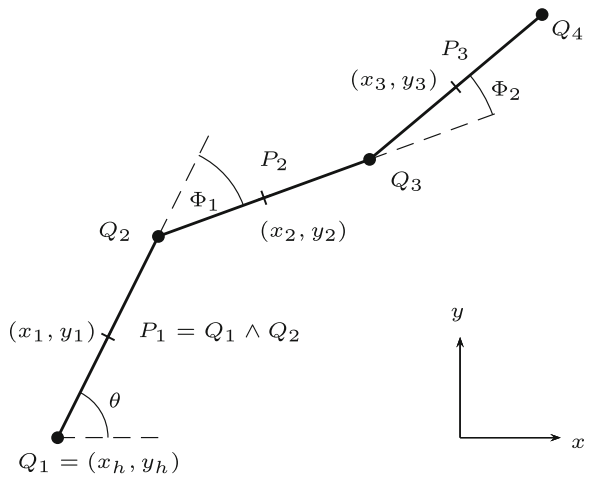


FIGURE 1. Snake robot model

and, for the robotic snake's initial position $x = y = \theta = \Phi_1 = \Phi_2 = 0$, three appropriate point pairs are established as

$$P_{1,0} = (e_0) \wedge (2e_1 + 2e_\infty + e_0) = 2e_0e_1 - 2e_+e_-,$$
$$P_{2,0} = (2e_1 + 2e_\infty + e_0) \wedge (4e_1 + 8e_\infty + e_0) = 2e_0e_1 + 8e_1e_\infty - 6e_+e_-,$$
$$P_{3,0} = (4e_1 + 8e_\infty + e_0) \wedge (6e_1 + 18e_\infty + e_0) = 2e_0e_1 + 24e_1e_\infty - 10e_+e_-.$$

Now, the transformations corresponding to the generalized coordinates can be written as

$$T_{x,y} = 1 - \frac{1}{2}(xe_1 + ye_2)e_\infty, \quad T_{Q_1} = 1 - \frac{1}{2}Q_1e_\infty, \quad T_{Q_2} = 1 - \frac{1}{2}Q_2e_\infty,$$
$$R_\theta = \cos\frac{\theta}{2} - L_0 \sin\frac{\theta}{2}, \quad \text{where } L_0 = T_{x,y}e_1e_2\tilde{T}_{x,y},$$
$$R_{\Phi_1} = \cos\frac{\Phi_1}{2} - L_1 \sin\frac{\Phi_1}{2}, \quad \text{where } L_1 = T_{Q_2}e_1e_2\tilde{T}_{Q_2},$$
$$R_{\Phi_2} = \cos\frac{\Phi_2}{2} - L_2 \sin\frac{\Phi_2}{2}, \quad \text{where } L_2 = T_{Q_3}e_1e_2\tilde{T}_{Q_3}.$$

The direct kinematics for the snake robot is obtained similarly as the kinematics for serial robot arms [1]. In general, it is given by a succession of generalised rotations R_i and it is valid for all geometric objects, including point pairs. A point pair P in a general position is computed from its initial position P_0 as follows

$$P = \prod_{i=1}^{n} R_i P_0 \prod_{i=1}^{n} \tilde{R}_{n-i+1}.$$

Unlike the fixed serial robot arms, we allow R_i to be also a translation. We view translations as degenerate rotations. Then the differential kinematics is expressed by means of the total differential as follows

$$dP = \sum_{j=1}^{n} \partial_{q_j} \left(\prod_{i=1}^{n} R_i P_0 \prod_{i=1}^{n} \tilde{R}_{n-i+1} \right) dq_j.$$

Since both the translations and the rotations can be expressed as exponentials, and $dR = d(e^{-\frac{1}{2}qL}) = -\frac{1}{2}RLdq$, the straightforward computation leads to the following assertion, [3,9]:

$$dP = \sum_{j=1}^{n} [P \cdot L_j] dq_j.$$

If R_i is a translation, then the axis of rotation L_i is given by a linear combination of bivectors that contain e_∞. In our particular case, previous considerations lead to the following general expression of $\dot{p}_i$, where the range of the subscript i is determined by the number of links:

$$\begin{aligned}\dot{p_i} &= \partial_t(P_i e_\infty \tilde{P_i}) = \dot{P_i} e_\infty \tilde{P_i} + P_i e_\infty \dot{\tilde{P}}_i \\ &= \sum_{j=1}^{n} [P_i \cdot L_j] e_\infty \tilde{P_i} dq_j + P_i e_\infty \sum_{j=1}^{n} [\tilde{L_j} \cdot \tilde{P_i}] dq_j \\ &= \frac{1}{2} \sum_{j=1}^{n} \left(P_i L_j e_\infty \tilde{P_i} - L_j P_i e_\infty \tilde{P_i} + P_i e_\infty \tilde{L_j} \tilde{P_i} - P_i e_\infty \tilde{P_i} \tilde{L_j} \right) dq_j\end{aligned}$$

In the last line we used the definition of the scalar product and the fact that L_i contains only bivectors. This also implies that L_i always commutes with e_∞ (in the case of a translation the product vanishes), and that $\tilde{L}_i = -L_i$. Thus we get

$$\dot{p_i} = \frac{1}{2} \sum_{j=1}^{n} \left(-L_j P_i e_\infty \tilde{P_i} - P_i e_\infty \tilde{P_i} \tilde{L_j} \right) dq_j = \sum_{j=1}^{n} [p_i \cdot L_j] dq_j,$$

i.e. the same formula of the differential kinematics holds also for the link centers p_i. Concretely, we obtain the system

$$\begin{aligned}\dot{p_1} &= [p_1 \cdot e_1 e_\infty]\dot{x} + [p_1 \cdot e_2 e_\infty]\dot{y} + [p_1 \cdot L_0]\dot{\theta}, \\ \dot{p_2} &= [p_2 \cdot e_1 e_\infty]\dot{x} + [p_2 \cdot e_2 e_\infty]\dot{y} + [p_2 \cdot L_0]\dot{\theta} + [p_2 \cdot L_1]\dot{\Phi}_1, \\ \dot{p_3} &= [p_3 \cdot e_1 e_\infty]\dot{x} + [p_3 \cdot e_2 e_\infty]\dot{y} + [p_3 \cdot L_0]\dot{\theta} + [p_3 \cdot L_1]\dot{\Phi}_1 + [p_3 \cdot L_2]\dot{\Phi}_2\end{aligned}$$

which in the matrix notation is of the form

$$\dot{p} = J\dot{q}, \tag{3}$$

where q are our coordinates (1) and $J = (j_{kl})$ is a 3×5 matrix with the elements defined by

$$\begin{aligned}&j_{i1} = [p_i \cdot e_1 e_\infty],\ j_{i2} = [p_i \cdot e_2 e_\infty], \\ &j_{ik} = [p_i \cdot L_{k-3}] \text{ for } 3 \le k < 3 + i, \\ &j_{ik} = 0 \text{ for } 3 + i \le k.\end{aligned} \tag{4}$$

As the wheels do not slip to the side direction, the velocity constraint condition is satisfied for each link i and in terms of CGA can be written as

$$\dot{p_i} \wedge P_i \wedge e_\infty = 0. \tag{5}$$

Thus if we substitute (3) in (5), we obtain a system of linear ODEs, which has a simple Pfaff matrix form

$$A\dot{q} = 0, \tag{6}$$

where $A = (a_{ij})$ is a matrix with the elements defined by

$$a_{ik} = j_{ik} \wedge P_i \wedge e_\infty. \tag{7}$$

Note that the enteries of A are multiples of $e_1 e_2 e_+ e_-$. Taking the conjugate and multiplying with e_3, A can be considered simply as a matrix over the field of functions. For example, the solution of this system with respect to $\dot{\theta}$ parameterized by $\dot{x}, \dot{y}$, (i.e. $\dot{x} = t_1$ and $\dot{y} = t_2$) is of the form

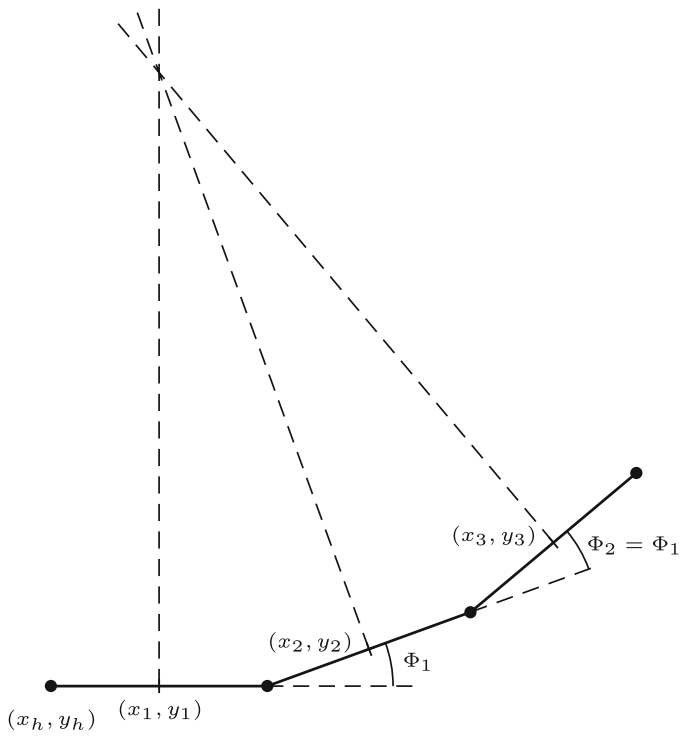


FIGURE 2. Snake robot's singular position

$$\dot{\theta} = -\frac{[p_1 \cdot e1e_\infty] \wedge P_1 \wedge e_\infty}{[p_1 \cdot L_0] \wedge P_1 \wedge e_\infty} t_1 - \frac{[p_1 \cdot e_2 e_\infty] \wedge P_1 \wedge e_\infty}{[p_1 \cdot L_0] \wedge P_1 \wedge e_\infty} t_2.$$

The straightforward computation leads to $[p_1 \cdot L_0] \wedge P_1 \wedge e_\infty = 2e_3^*$, i.e. the solution always exists, because $([p_1 \cdot L_0] \wedge P_1 \wedge e_\infty)^{-1} = -\frac{1}{2} e_3^*$. The system matrix is singular in case that the wheel axes, i.e. lines perpendicular to each link containing the link center point, intersect in precisely one point or are parallel, see Fig. 2.

In our setting this is one condition only because in CGA the parallel lines intersect in exactly one point which is e_∞. It is easy to see that this happens in such case that all joints lie on a single circle, i.e. in CGA they satisfy a simple condition

$$P_1 \wedge P_3 = 0. \tag{8}$$

Finally, note that the non-singular solution forms a 2-dimensional distribution which can be parameterized e.g. as follows:

$$\dot{q} = G \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}, \tag{9}$$

where $G = (g_{ij})$ is a 2×5 control matrix with the elements defined by

$$\begin{aligned}
&g_{11} = 1, \quad g_{12} = 0, \quad g_{21} = 0, \quad g_{22} = 1, \quad g_{31} = \cos(\theta), \quad g_{32} = \sin(\theta), \\
&g_{41} = -2\cos(\Phi_1)\sin(\theta) + \sin(\theta + \Phi_1) - \sin(\theta), \\
&g_{42} = 2\cos(\Phi_1)\cos(\theta) - \cos(\theta + \Phi_1) + \cos(\theta), \\
&g_{51} = 4\cos(\Phi_1)\cos(\Phi_2)\sin(\theta) - 2\sin(\theta + \Phi_1)\cos(\Phi_2) + 2\cos(\Phi_1)\sin(\theta) \\
&\qquad - 2\cos(\Phi_1 + \Phi_2)\sin(\theta) - \sin(\theta + \Phi_1) + \sin(\theta + \Phi_1 + \Phi_2),
\end{aligned}$$

$$\begin{aligned} g_{52} = &-4\cos(\Phi_1)\cos(\Phi_2)\cos(\theta) + 2\cos(\theta+\Phi_1)\cos(\Phi_2) \\ &- 2\cos(\Phi_1)\cos(\theta) + 2\cos(\Phi_1+\Phi_2)\cos(\theta) + \cos(\theta+\Phi_1) \\ &- \cos(\theta+\Phi_1+\Phi_2). \end{aligned}$$

Thus if we consider the snake robot's configuration space with coordinates (1) as a 5-dimensional manifold M, the solution above forms a couple of vector fields g_1 and g_2.

It is clear, that the space $\operatorname{span}\{g_1, g_2\}$ determines the set of accessible velocity vectors and thus, taking into account the vector field flows $\exp(tg_1)$, $\exp(tg_2)$, the possible trajectories of the snake robot. On the other hand, due to non-commutativity of $\exp(tg_1)$, $\exp(tg_2)$, the robot can move even along the flow of the Lie bracket by means of the composition

$$\exp(-tg_2) \circ \exp(-tg_1) \circ \exp(tg_2) \circ \exp(tg_1).$$

Extending this idea, the space Q_q of all movement directions in point q is given by taking all possible Lie brackets of $g_1(q)$ and $g_2(q)$ and the resulting vector fields. From the geometric control theory point of view, it is quite necessary that the dimension of Q_q is equal to the dimension of the tangent space $T_qM, q \in M$, which in our case is 5. Note that this is the condition on the model local controllability given by the Rashevsky–Chow Theorem. In our case, it is easy to show that in regular points q indeed

$$Q_q = \operatorname{span}\{g_1, g_2, [g_1, g_2], [g_1, [g_1, g_2]], [g_2, [g_1, g_2]]\} \cong T_qM.$$

Thus the tangent space to the configuration space of the snake robot is equipped with a $(2, 3, 5)$ filtration.

4. CLUCalc Implementation

The proposed snake control was tested in CLUCalc software [2,7], which is designed exactly for calculations in arbitrary predefined geometric algebra. The following code piece contains the definitions of basic objects:

```
DefVarsN3();\\
// COMPUTATION OF COORDINATES IN CONFIGURATION SPACE\\
// POSITION OF ROBOT CORR. TO THE COORDINATES\\
// Initial position of points Q1, Q2, Q3 and line L1 \\
Q1=VecN3(0,0,0);
L1=VecN3(0,0,1);
T0=TranslatorN3(2,0,0);
Q2=T0*P1*~T0;
Q3=T0*P2*~T0;
Q4=T0*P3*~T0;
// Computation of point pairs P1,P2 and P2
P1=P1^P2;
P2=P2^P3;
P3=P3^P4;
```

The initial position is thus recalculated with respect to the controlling parameters change.

```
\\ Coordinates x and y from configuration space
T=TranslatorN3(x,y,0);
\\ Axis of rotation in point (x,y)
LB=T*L1*~T;
\\ Rotor in space with respect to R
MB=TranslatorN3(LB)*RotorN3(0,0,1,d)*~TranslatorN3(LB);
\\ New position of point pair P1
:P1=MB*T*P1*~T*~MB;
\\ Projection to the first point of P1
T1=(-(sqrt(P1.P1)+P1)/(einf.P1));
\\ Rotor in first joint
L2=TranslatorN3(T1)*L1*~TranslatorN3(T1);
M1=TranslatorN3(L2)*RotorN3(0,0,1,a)*~TranslatorN3(L2);
\\ New position of point pair P2
:P2=M1*MB*T*P2*~T*~MB*~M1;
\\ Projection to the second point of P3
T2=((sqrt(P3.P3)+P3)/(einf.P3));
\\ Rotor in second joint
L3=TranslatorN3(T2)*L1*~TranslatorN3(T2);
M2=TranslatorN3(L3)*RotorN3(0,0,1,b)*~TranslatorN3(L3);
\\ New position of point pair P3
:P3=M2*M1*MB*T*P3*~T*~MB*~M1*~M2;
```

Figure 3 demonstrates the evolution from 0 in the direction of the vector field g_1, i.e. when the controlling parameter t_1 is set to zero and t_2 is changed within the range $\langle 0, 2\pi \rangle$.

Figure 4 shows the motion corresponding to the bracket $[g_1, g_2]$ which is realized by means of a periodic transformation of the generators g_1 a g_2:

$$v(t) = -\epsilon\omega \sin(\omega t) g_1 + \epsilon\omega \cos(\omega t) g_2,$$

where $\epsilon = 0.1$, $\omega = 4$ and $t \in \langle 0, 2\pi/\omega \rangle$.

The calculations with the CGA elements provide not only significant theoretical model simplification, particularly in better geometric understanding and the notations, but also a remarkable computational complexity reduction when a system such as Gaalop is used [2]. To demonstrate the contribution of CGA calculus let us restrict from CGA to quaternions. Classically, rotations in 3D-space are realized by multiplication of a 3x3 matrix. The composition of rotations is then rotation matrices multiplication, i.e such composition is given by 27 multiplications and 18 summations. If you represent rotations by quaternions, for the same composition you need just 16 multiplications and 15 summations. This reduces the computational time remarkably.

FIGURE 3. $t_1 = 0, t_2 = t$ (pictured by CLUCalc)

FIGURE 4. Lie bracket $[g_1, g_2]$ (pictured by CLUCalc)

5. Dynamics

The results obtained above via the conformal geometric algebra may be used directly to derive the model of snake's dynamics from the d'Alembert principle. Since there is no potential energy, the Lagrangian $\mathcal{L}$ of the (unconstrained) system equals to the total kinetic energy

$$\mathcal{L} = E_k. \tag{10}$$

Then, the Euler–Lagrange equation has the form

$$\frac{d}{dt}\left(\frac{\partial E_k}{\partial \dot{q}}\right) - \frac{\partial E_k}{\partial q} = F + \tau. \tag{11}$$

The right-hand side of the Eq. (11) contains the traction forces F and the control forces (torques) $\tau = (0, 0, 0, \tau_1, \tau_2)^T$. Since the traction forces are always perpendicular to the evolution of the system, we have $F = \lambda A^T$, where A is the matrix that gives the Pfaff constraint (6), viewed as a 3×5 matrix over the field of functions. And since this matrix satisfies $A(q)G(q) = 0$ in each point q, we eliminate the traction forces term by pre-multiplying (11) by 2×5 matrix G^T. The resulting system together with (6) then describes the dynamics of the snake robot.

In order to get a precise form of the equations, it is sufficient to compute the total kinetic energy E_k. We consider a snake with identical links and wheels, where the wheels are located in centers of links. If we denote by m_w the mass of each wheel, r_w and d_w its radius and thickness, m_l the mass of each link and $L = 2$ its length, the moment of inertia of each link with the wheel (with respect to the axis located in its center of mass) is given by

$$I = \frac{1}{12} m_w \left(3r_w^2 + d_w^2\right) + \frac{1}{12} m_l L^2.$$

Denoting by $m = m_w + m_l$ the total mass of wheel and link, we compute the kinetic energy of the individual links as

$$\begin{aligned} E_{k1} &= \frac{1}{2} m \dot{p}_1^2 + \frac{1}{2} I \dot{\theta}^2, \\ E_{k2} &= \frac{1}{2} m \dot{p}_2^2 + \frac{1}{2} I (\dot{\theta} + \dot{\phi}_1)^2, \\ E_{k3} &= \frac{1}{2} m \dot{p}_3^2 + \frac{1}{2} I (\dot{\theta} + \dot{\phi}_1 + \dot{\phi}_2)^2. \end{aligned}$$

Recall that p_i is the position of the i-th link center (that coincides with its mass center) and $\dot{p}_i$ represents its velocity which is computed using the equation of differential kinematics (3) explained in the previous section. Although $\dot{p}_i$ is a vector in CGA, its square (with respect to the geometric product) is a real number which gives the square of the real (physical) velocity. Hence the total energy of the snake robot

$$E_k = \frac{1}{2}m\left(\dot{p}_1^2 + \dot{p}_2^2 + \dot{p}_3^2\right) + \frac{1}{2}I\left(\dot{\theta}^2 + (\dot{\theta}+\dot{\phi}_1)^2 + (\dot{\theta}+\dot{\phi}_1+\dot{\phi}_2)^2\right)$$

can be written in a compact way as

$$E_k = \frac{1}{2}\dot{q}^T\mathcal{M}\dot{q}, \tag{12}$$

where $\mathcal{M}$ is a symmetric 5×5 matrix

$$\mathcal{M} = mJ^TJ + I\begin{pmatrix} 0 & \multicolumn{3}{c}{0} \\ \hline & 3 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ & 1 & 1 & 1 \end{pmatrix}. \tag{13}$$

Recall that J is a 3×5 matrix given by (4). Now, using (12) we directly compute

$$\frac{\partial E_k}{\partial \dot{q}} = \mathcal{M}\dot{q}$$

and

$$\frac{\partial E_k}{\partial q} = \frac{1}{2}\dot{q}^T\left(\frac{\partial \mathcal{M}}{\partial q}\right)\dot{q}.$$

Thus the Lagrange Eq. (11) can be rewritten as

$$\mathcal{M}\ddot{q} - \dot{\mathcal{M}}\dot{q} - \frac{1}{2}\dot{q}^T\left(\frac{\partial \mathcal{M}}{\partial q}\right)\dot{q} = \lambda A^T + \tau.$$

A standard elimination of the multipliers then results in the following equations of motion of the snake robot

$$\begin{aligned} A\dot{q} &= 0, \\ \left(G^T\mathcal{M}\right)\ddot{q} - G^T\left(\dot{\mathcal{M}} + \frac{1}{2}\dot{q}^T\frac{\partial \mathcal{M}}{\partial q}\right)\dot{q} &= G^T\tau. \end{aligned} \tag{14}$$

Recall that A is given by (7), $\mathcal{M}$ is given by (13) and G is the control matrix (9). Thus the equations of motion are formed by three first order differential equations and two second order differential equations. For more informations one can see [10], for example.

6. Conclusions

The snake-like robot (see [5]) modelling by means of CGA is quite promising topic. This approach leads to a list of nice formulas for forward kinematics (2), differential kinematics (3), nonholonomic constrains (5) and singular positions (8). It brings not only a geometric point of view to the topic that is important for the complete problem understanding, but it also allows the

calculations to be done faster and thus it contributes to the final model realization together with the proposal of the control solution.

References

[1] Gonzalez-Jimenez, L., Carbajal-Espinosa, O., Loukianov, A., Bayro-Corrochano, E.: Robust Pose Control of Robot Manipulators Using Conformal Geometric Algebra. Adv. Appl. Clifford Algebr. **24**(2), 533–552 (2014)

[2] Hildenbrand, D.: Foundations of Geometric Algebra Computing. Springer, Heidelberg (2013)

[3] Hrdina, J., Vašík, P.: Notes on differential kinematics in conformal geometric algebra approach. In: Matoušek, R. (ed.) Mendel 2015, Advances in Intelligent Systems and Computing, vol. 378, pp. 375–385. Springer (2015)

[4] Liljebäck, P., Pettersen, K.Y., Stavdahl, Ø., Gravdahl, J.T.: Snake Robots, Modelling, Mechatronics and Control. Springer, London (2013)

[5] Matoušek, R., Návrat, A.: Trident snake control based on CGA. In: Matoušek, R. (ed.) Mendel 2015, Advances in Intelligent Systems and Computing, vol. 378, pp. 375–385. Springer, Heidelberg (2015)

[6] Murray, R.M., Zexiang, L., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton (1994)

[7] Perwass, Ch.: Geometric Algebra with Applications in Engineering. Springer, Heidelberg (2009)

[8] Selig, J.M.: Geometric Fundamentals of Robotics. Springer, New York (2004)

[9] Zamora-Esquivel, J., Bayro-Corrochano, E.: Kinematics and differential kinematics of binocular robot heads. In: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, pp. 4130–4135. IEEE, Orlando, FL (2006)

[10] Zamora-Esquivel, J., Bayro-Corrochano, E.: Parallel Forward Dynamics: a geometric approach. In: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2377–2382. IEEE, Taipei (2010)

Jaroslav Hrdina, Aleš Návrat and Petr Vašík
Institute of Mathematics
Faculty of Mechanical Engineering
Brno University of Technology
Brno
Czech Republic
e-mail: hrdina@fme.vutbr.cz

Aleš Návrat
e-mail: navrat.a@fme.vutbr.cz

Petr Vašík
e-mail: vasik@fme.vutbr.cz

Received: January 31, 2015.
Accepted: October 23, 2015.

Appendix 4

[23] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: *CGA-based robotic snake control,*, Adv.Appl. Clifford Algebras, (in print), 1–12 online, 2016.

Adv. Appl. Clifford Algebras

DOI 10.1007/s00006-016-0695-5

Advances in
Applied Clifford Algebras

CGA-based robotic snake control

Jaroslav Hrdina*, Aleš Návrat, Petr Vašík and R. Matoušek

Abstract. Local controllability of an n-link robotic snake with variable wheel positions is solved by means of the conformal geometric algebra, more precisely by the Clifford algebra of signature $(3, 1)$. The nonholonomic kinematic equations are assembled, their role in the geometric control theory is discussed and the singular positions are elaborated. Within this paper, we present an alternative model description only, while all its kinematic properties remain.

Mathematics Subject Classification. Primary 93C10; Secondary 15A66.

Keywords. Conformal geometric algebra, Clifford algebra, Mathematical robotics, Nonholonomic mechanics, Snake robots, Local controllability, Bionics.

1. Introduction

Within this paper, we consider an n-link snake robot moving on a planar surface. More precisely, it is a model when to each link, two wheels are attached and thus the possible movement directions are determined uniquely. Calculations in CGA allow the wheels to be placed not in the link's centre of mass only, but their position is arbitrary within each link. The aim is to find the complete kinematic description. Although we handle only the case that the links are of a constant length 1, the generalization to an arbitrary length of each link is obvious. If the generalized coordinates are considered, the non-holonomic forward kinematic equations can be understood as a Pfaff system. In classical approach, local controllability is discussed by means of differential geometry and Lie algebras, see [12,15]. Our aim is to translate the whole kinematics into the language of CGA, where both linear objects and spheres of dimensions 2, 1 and 0, see [3,4,14,16], are easy to transform.

The classical approach composes the kinematic chain of homogeneous matrices using the moving frame method and Euler angles [10]. Instead of

The authors were supported by a Grant No. FSI-S-14-2290.
*Corresponding author.

Published online: 09 June 2016

this, our aim is to use the notions of conformal geometric algebra. Particularly, we use a Clifford algebra $\mathcal{C}l(3,1)$ where the Euclidean space $\mathbb{E}_2$ is included by a mapping $x \mapsto x + \frac{1}{2}x^2 e_\infty + e_0$.

In particular, the point pair is used to derive the kinematic equations and for the control of the robotic snake, consequently. More precisely, to any link of a snake a single point pair is assigned and the mechanism is transformed by rotations and translations. We introduce the differential kinematic equations, as well as the non-holonomic conditions, respectively. Also the singularity conditions are formulated. The advantage of CGA description lies in the simplicity of the model modifications, i.e. variable link length and variable wheel position. Furthermore, we fully use the advantage of CGA in operations representation, precisely rotations and translations are represented by particular CGA elements.

Currently, the model of a snake robot is well elaborated by classical geometrical approaches [10], together with the controllability and motion control. Also heuristic algorithms for the control are known. On the other hand, the CGA description has not been presented as far as the authors are aware. Note that the solutions of local controllability, motion control model and singular points analysis can be adopted from the classical models, see e.g. [10]. Within this paper, we present an alternative model description only, while all its kinematic properties remain.

2. State of the art

The topic of the snake-like robots goes back to early 1970's when Hirose formulated the essential model design and developed limbless locomotors, for the complex review of his work see [7]. He started the first bio-mechanical study using the real snakes and designed the first snake-like robot based on so-called serpentine locomotion.

The first designs of Hirose's snake robots had modules with small passive wheels, and since then, most of the current developments by Downling [2], Chirikjian and Burdick [1], and Ostrowski [13] keep using the snake robots with wheels in order to facilitate forward propulsion.

Since then a variety of different snake-like robots has been designed For the precise modelling of snake-like motions see [10]. For the snake-like motion mimicking, basically three approaches are used (simplified classification):

(1) The original Hirose approach [7], i.e. the representation of the kinematic model based on the serpenoid curve. This is well developed in many directions. Two main planar models, the serpentine locomotion (horizontal locomotion) and the travelling wave locomotion (vertical locomotion) are often adopted. Furthermore, many authors present the optimized versions of this approach based on various parameter estimations. This is also often used for the verification of other solutions.
(2) Heuristic and artificial intelligence approach based on so-called CPG (Central Pattern Generators) or other types of non-linear models. CPG produces so called rhythmic outputs defining a periodic motion of a

robotic snake. It is realized as the intra-spinal artificial neural network. It is one of the mechanisms for neural control of locomotion in higher animals and humans.

(3) The analytical mechanic model based on the Newton–Euler formulation and Lagrangian [12,15]. The fact that the energy of the system and the frictional forces acting on the system are invariant with respect to the position and orientation of the snake robot is exploited to simplify the mathematical model. In the position calculation, the classical Euler angles are used.

Note that the above approaches differ not only by mathematical methods but also by suitability for particular applications. The division into global and local control is vital. While heuristic methods suit well to the global control, the analytical approach (3) solves rather local control, i.e. the way of actual movements. Furthermore, while the areas (1) and (2) are ready to be used, there is still plenty open questions, such as the optimality and/or effectiveness, and space for possible improvements in (3). In this sense, analytical description by CGA seems to be promising.

3. Conformal geometric algebra: CGA

We recall some elementary facts about CGA and specify our particular setting. Note that the properties and definitions of conformal geometric algebras can be found in e.g. [12]. Classically, for modelling a 3D robot, the whole CGA (i.e. $\mathcal{C}l(4,1)$) is used, where the embedding $\mathbb{R}^3 \to \mathbb{R}^{4,1}$ is considered. As the snake robot moves on a planar surface it is enough to consider an embedding

$$c : \mathbb{R}^2 \to \mathbb{K}^3 \subset \mathbb{R}^{3,1},$$

where $\mathbb{K}^3$ is a null cone and $\mathbb{R}^{3,1}$ is a Minkowski space. Let us choose the basis of $\mathbb{R}^{3,1}$ such that the appropriate quadratic form is

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix},$$

which corresponds to the choice of the basis $e_1, e_2, e_0, e_\infty \in \mathbb{R}^{3,1}$ composed of the vectors from the classically used basis of $\mathbb{R}^{4,1}$. Consequently, the embedding is of the form

$$c(x) = x + \frac{1}{2}x^2 e_\infty + e_0.$$

Note that e_0 and e_∞ play the role of the origin and the infinity, respectively.

We recall that the algebra operation is called the geometric product and two further elementary operations on CGA are added. We recall just the basic properties used within the following text. The *inner (dot) product* and *outer (wedge) product* on $\mathbb{R}^{3,1}$ correspond to the symmetric and antisymmetric part

of the geometric product, respectively. Generally, the wedge product of two basis blades E_k and E_l of grades k and l, respectively, is defined as

$$E_k \wedge E_l := \langle E_k E_l \rangle_{k+l}$$

and the dot product is defined as

$$E_k \cdot E_l := \langle E_k E_l \rangle_{|k-l|},$$

where $\langle \ \ \rangle_k$ is the grade projection into grade k. Consequently, the inverse mapping

$$c^{-1} : \mathbb{K}^3 \to \mathbb{R}^2$$

is defined by

$$c^{-1}(x) = P^{\perp}_{e_\infty \wedge e_0} \left(\frac{x}{-x \cdot e_\infty} \right) = x + \frac{x \cdot (e_\infty \wedge e_0)^{-1}}{x \cdot e_\infty} (e_\infty \wedge e_0),$$

where $P^{\perp}$ denotes the orthogonal complement to the projection onto $e_\infty \wedge e_0$.

Algebraically, we consider the geometric algebra on $\mathbb{R}^{3,1}$, i.e. the algebra denoted as $\mathbb{G}_{3,1}$ which is the Clifford algebra $\mathcal{C}l(3,1)$. The advantage of representing objects in CGA are well known. Particularly, not only spheres of all dimensions but also transformations (rotations and translations) are represented as algebra elements and this provides high efficiency in calculations. For instance, a sphere of any dimension (point, circle, point pair, line) $S \subseteq \mathbb{R}^2$ is viewed as an element $A \in \mathbb{G}_{3,1}$ such that

$$x \in S \Leftrightarrow c(x) \wedge A = 0.$$

Note that this is so-called OPNS representation. Dually, i.e. in IPNS representation, the sphere S is represented by an element A^* and the condition reads

$$x \in S \Leftrightarrow c(x) \cdot A^* = 0.$$

The transformation represented by T of a sphere A is in CGA realized by conjugation

$$A \mapsto T A \tilde{T}.$$

For instance, the translation in the direction $t = t_1 e_1 + t_2 e_2$ is realized by the multivector

$$T = 1 - \frac{1}{2} t e_\infty$$

and the rotation around the origin by angle ϕ is realized by the multivector

$$R = \mathrm{e}^{-\frac{1}{2}(e_1 \wedge e_2) L} = \cos \frac{\phi}{2} - (e_1 \wedge e_2) \sin \frac{\phi}{2}.$$

The rotation around a general point is then a composition

$$T R \tilde{T}$$

of the translation to the origin, rotation R and reverse translation.

4. Snake robot model

The snake robot described in this paper consists of n rigid links interconnected by $n-1$ motorized joints. To each line, a pair of wheels is attached to provide an important snake-like property that the ground friction in the direction perpendicular to the link is considerably higher than the friction of a simple forward move. In particular, this prevents the slipping sideways. To describe the actual position of a snake robot the generalized coordinates

$$q = (x, y, \theta, \Phi_1, \ldots, \Phi_{n-1}) \tag{1}$$

are considered, see Fig. 1.

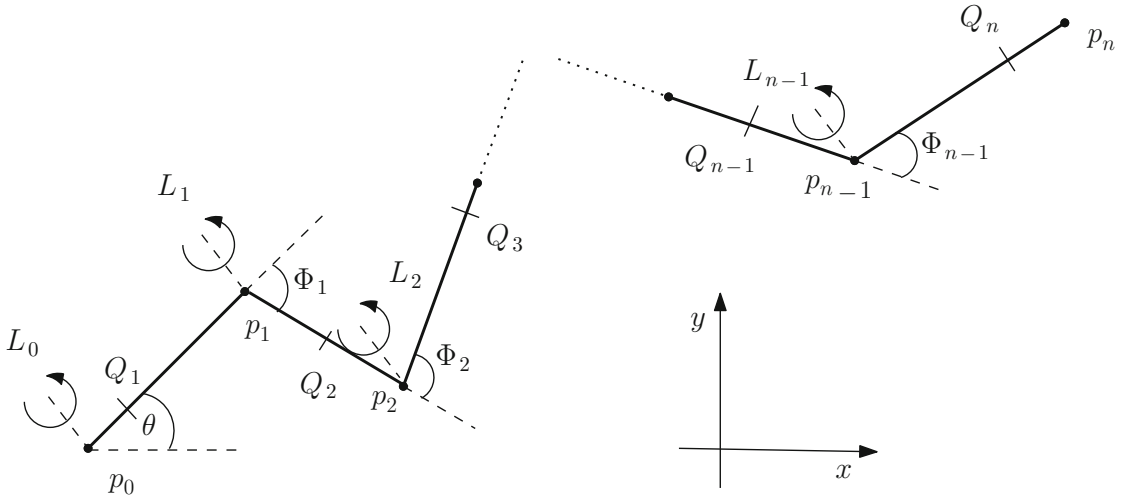


FIGURE 1. Snake robot model

Note that a fixed coordinate system (x, y) is attached. For sake of simplicity, we consider the links to be of constant length 1 but the generalization to arbitrary lengths is obvious. The points $p_i := (x_i, y_i), i = 0, \ldots, n$, denote the endpoints of each link and by $Q_i = r_i p_i + (1 - r_i)p_{i-1},\ r_i \in \langle 0, 1\rangle,\ i = 1, \ldots, n$, we denote the points where the wheels are attached to the particular link. Then, the distance $|Q_i p_{i-1}| = -2(Q_i \cdot p_{i-1})$ is equal to r_i. If the absolute angle of the ith link, i.e. the angle between the link and the x-axis, is denoted by θ_i then the position of Q_i w.r.t. the global $x - y$ axes is then expressed as

$$\begin{aligned} Q_{x,i} &= P_{x,0} + \textstyle\sum_{j=1}^{i-1} \cos\theta_j + r_i \cos\theta_i, \\ Q_{y,i} &= P_{y,0} + \textstyle\sum_{j=1}^{i-1} \sin\theta_j + r_i \sin\theta_i. \end{aligned} \tag{2}$$

Note that to recover the generalized coordinates one has to consider the assertion

$$\theta_i = \sum_{j=1}^{i-1} \Phi_j + \theta.$$

Furthermore, the linear velocity of Q_i can be determined by taking the derivative of (2) and thus the nonholonomic equations are obtained. This gives

us a rough idea of the snake model in the Euclidean space $\mathbb{R}^{n+2}$. To describe the robotic snake by means of CGA we use as a central object the point pairs

$$P_i = p_{i-1} \wedge p_i, \qquad i = 1, \dots, n$$

and thus the i-th link is represented by a point pair P_i. Anyway, if the position of a particular joint p_i is needed, one can consider the projection of a point pair onto its endpoints in the form

$$p_{i-1} = \frac{-\sqrt{P_i \cdot P_i} + P_i}{e_\infty \cdot P_i}, \quad p_i = \frac{\sqrt{P_i \cdot P_i} + P_i}{e_\infty \cdot P_i}.$$

The direct kinematics for the snake robot is obtained similarly as the kinematics for serial robot arms [16]. For the case of a 3-link robotic snake see [9]. In general, it is given by a succession of generalised rotations R_i and translations T_i. The composition of R_i and T_i will be called a motor and will be denoted by M_i. Particularly, the actual position of a joint p_i at a general point $q = (x, y, \theta, \Phi_1, \dots, \Phi_{n-1}) \in \mathbb{R}^{n+2}$ is computed from its initial position $p_i(0)$ by

$$p_i(q) = M_i p_i(0) \tilde{M}_i, \qquad i = 0, \dots, n,$$

where $p_i(0)$ is the initial position of p_i and M_i is a motor defined as

$$M_0 := T = 1 - \frac{1}{2}(xe_1 + ye_2)e_\infty,$$

i.e. T stands for the translation from the origin to the position of the head point, and

$$M_i = R_i \dots R_1 T \qquad \text{for } i \neq 0,$$

i.e. the product of rotations. These can be determined inductively as follows:

$$R_{i+1} = \mathrm{e}^{-\Phi_i L_i} = \cos\frac{\Phi_i}{2} - \sin\frac{\Phi_i}{2} L_i,$$
$$L_i = M_i L_i(0) \tilde{M}_i,$$

where L_i are the axes of rotations placed in the corresponding joints, see Fig. 1. Furthermore, the wheel position at the link P_i is calculated as

$$Q_i = M_i Q_i(0) \tilde{M}_i. \tag{3}$$

In the following, the snake robot's initial position is the one depicted in Fig. 2, i.e $q_0 = (0, \dots, 0)$. Then

$$p_i(0) = ie_1 + \frac{1}{2} i^2 e_\infty + e_0,$$
$$L_i(0) = ie_2 \wedge e_\infty - e_1 \wedge e_2.$$

This gives us the whole kinematic chain which corresponds to Eq. (2). The differential kinematics will be obtained by the differentiation of this kinematic chain as follows. For the wheel position point $Q_i \in P_i$ we have

$$\dot{Q}_i = Q_i \cdot (e_1 \wedge e_\infty)\dot{x} + Q_i \cdot (e_2 \wedge e_\infty)\dot{y} + \sum_{j=0}^{i-1} (Q_i \cdot L_j)\dot{\Phi}_j. \tag{4}$$

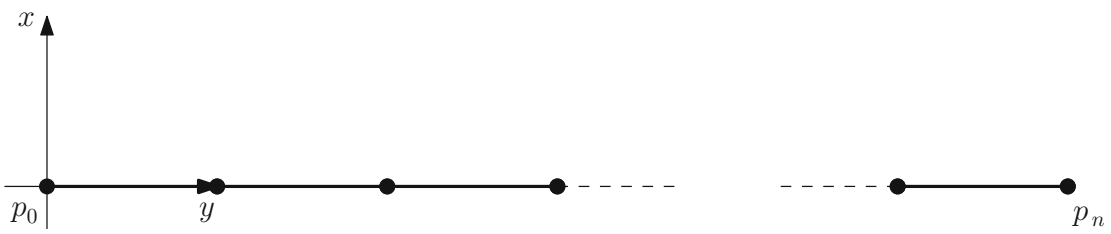


FIGURE 2. Snake robot initial position

Note that according to [8], the Eq. (4) holds for any other point on the link P_i and, with minor modification, for any CGA object attached to P_i at the position of Q_i .

If we consider the wheels positions as a vector $Q = (Q_i)^T$, the Eq. (4) transforms as

$$\dot{Q} = J\dot{q}, \tag{5}$$

where

$$J = \big(\underbrace{Q_i \cdot (e_1 \wedge e_\infty)\; Q_i \cdot (e_2 \wedge e_\infty)}_{n\times 2} \;\big|\; \underbrace{Q_i \cdot L_{j-1}}_{n\times n}\big), \quad i,j = 1,\dots,n,$$

plays role of a Jacobi matrix, particularly a matrix of inner products of Q_i and axes of rotations or translations.

As the wheels do not slip to the side direction, the velocity vector must be parallel to $\dot{Q}_i$ and the constraint condition, i.e. the nonholonomic constraint, is in terms of CGA expressed as

$$\dot{Q}_i \wedge P_i \wedge e_\infty = 0. \tag{6}$$

Thus if we substitute (4) in (6), we obtain a system of linear ODEs

$$A\dot{q} = 0, \tag{7}$$

where $A = (a_{ij})$ plays the role of the Pfaff matrix and is of a simple form

$$a_{ij} = J_{ij} \wedge P_i \wedge e_\infty. \tag{8}$$

Note that the elements of A are just pseudoscalar multiples and thus A can be understood as a matrix over the field of functions. To specify the elements of A more precisely we formulate the following Lemma. For sake of simplicity, we suppose that each link length is 1 and the wheels attached to the ith link are represented by a point Q_i but both these parameters can be easily generalized. In particular, each link can be of a different length and Q_i can stand for any CGA object.

Lemma 4.1. *If $Q_i = r_i p_i + (1 - r_i)p_{i-1}$, $r_i \in \langle 0, 1\rangle$, is a point on the link P_i, then*

$$(Q_i \cdot L_{i-1}) \wedge P_i \wedge e_\infty = r_i I, \tag{9}$$

where I is a pseudoscalar.

Proof. First, let us note that the rotor R_i commutes w.r.t. to the geometric product with its appropriate axis of rotation L_{i-1}. This follows from the fact that

$$R_i = \mathrm{e}^{-\Phi_{i-1}L_{i-1}} = \sum_{k=0}^{\infty} \frac{1}{k!}\Phi_{i-1}^k \underbrace{L_{i-1}\dots L_{i-1}}_{k}.$$

Then

$$\begin{aligned} Q_i \cdot L_{i-1} &= (R_i M_{i-1} Q_i(0) \tilde{M}_{i-1} \tilde{R}_i) \cdot L_{i-1} \\ &= \langle R_i M_{i-1} Q_i(0) \underbrace{\tilde{M}_{i-1} M_{i-1}}_{1} L_{i-1}(0) \tilde{M}_{i-1} \tilde{R}_i \rangle_1 \\ &= M_i (Q_i(0) \cdot L_{i-1}(0)) \tilde{M}_i \end{aligned}$$

and thus

$$(Q_i \cdot L_{i-1}) \wedge P_i \wedge e_\infty = M_i((Q_i(0) \cdot L_{i-1}(0)) \wedge P_i(0) \wedge e_\infty)\tilde{M}_i$$

and because $(Q_i(0) \cdot L_{i-1}(0)) \wedge P_i(0) \wedge e_\infty$ is a multiple of a pseudoscalar, we have

$$(Q_i \cdot L_{i-1}) \wedge P_i \wedge e_\infty = (Q_i(0) \cdot L_{i-1}(0)) \wedge P_i(0) \wedge e_\infty, \tag{10}$$

i.e. again a constant multiple of a pseudoscalar. If we consider the initial position to be the zero position as in Figure 2 and substitute it into (10) we obtain

$$(Q_i \cdot L_{i-1}) \wedge P_i \wedge e_\infty = r_i I$$

for $r_i \in \langle 0, 1 \rangle$ and a pseudoscalar I. □

If we denote the element $e_i \wedge e_\infty$ by $e_{i\infty}$ then Lemma 4.1 directly implies the following

Proposition 4.2. *The Pfaff matrix A of the system (7) is of the form*

$$A = (b|B),$$

where

$$b = \begin{pmatrix} e_{1\infty} \wedge P_1 & e_{2\infty} \wedge P_1 \\ e_{1\infty} \wedge P_2 & e_{2\infty} \wedge P_2 \\ \vdots & \vdots \\ e_{1\infty} \wedge P_n & e_{2\infty} \wedge P_n \end{pmatrix}$$

and

$$B = \begin{pmatrix} r_1 & 0 & \cdots & 0 \\ (Q_2 \cdot L_0) \wedge P_2 \wedge e_\infty & r_2 & & \\ \vdots & & \ddots & 0 \\ (Q_n \cdot L_0) \wedge P_n \wedge e_\infty & \cdots & (Q_n \cdot L_{n-2}) \wedge P_n \wedge e_\infty & r_n \end{pmatrix}.$$

One can see that the matrix B in the form from Proposition 4.2 is a lower triangle block matrix with nonzero elements on the diagonal (and thus always invertible). The inverse of such matrix is easy to express. Thus generally, the matrix $A = (b|B)$, where $b = (b_{ij})$, $i = 1, \ldots, n$, $j = 1, 2$, and $B = (B_{ij}), i, j = 1, \ldots, n$, are $n \times 2$ and $n \times n$ dimensional matrices, respectively, and their elements can be expressed as

$$\begin{aligned} b_{ij} &= (Q_i \cdot e_{j\infty}) \wedge P_i \wedge e_\infty = e_{j\infty} \wedge P_i, \\ B_{ij} &= (Q_i \cdot L_{j-1}) \wedge P_i \wedge e_\infty. \end{aligned} \tag{11}$$

Thus if we consider the control $u = (u_1, u_2)$, $u_1 = \dot{x}$, $u_2 = \dot{y}$ then the control matrix G is in the form

$$G = \begin{pmatrix} E \\ -B^{-1}b \end{pmatrix},$$

wher E is a 2×2 unit matrix, the system (5) can be written as

$$\dot{Q} = J\dot{q} = JGu,$$

where

$$JG = \big(\underbrace{Q \cdot e_{1\infty}\ Q \cdot e_{2\infty}}_{n\times 2} \big) - \big(\underbrace{Q \cdot L_{j-1}}_{n\times n} \big) \underbrace{B^{-1}b}_{n\times 2}.$$

Moreover, once the model is reformulated in this sense, the velocity equations of the wheel points change accordingly, e.g. the equation of the wheel point Q_n attached to the last link P_n will be of the form

$$\dot{Q}_n = Q_n \cdot (\ell_1\ \ell_2) \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},$$

where

$$\begin{aligned} \ell_1 &= e_{1\infty} - \sum_{i=1}^{n} L_{i-1} \sum_{j=1}^{n} B_{ij}^{-1} b_{j1} \\ \ell_2 &= e_{2\infty} - \sum_{i=1}^{n} L_{i-1} \sum_{j=1}^{n} B_{ij}^{-1} b_{j2} \end{aligned}$$

provided that B_{ij} and b_{ij} are the elements of B and b, respectively, specified by (11). This completes the CGA-based model control description.

5. Singular points

The singular positions, i.e. singular points in $\mathbb{R}^{n+2}$ in generalized coordinates, can be characterized as those positions that do not allow the snake-like motion without breaking the nonholonomic constraints. Note that the whole mechanism can move linearly (see the zero initial position in Fig. 2) or rotate around a given point without changing the Φ_i—coordinates. Yet these motions are not considered as snake-like as they are not a consequence of the mechanism construction but rather of the outer forces.

The singular point example is the following: a position is singular if all wheel axles o_i intersect in one point, see Fig. 3.

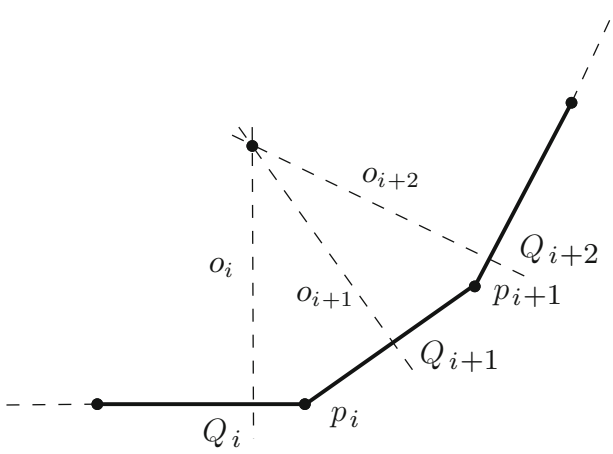


FIGURE 3. Snake robot's singular position

In CGA description, this condition is simply expressed as

$$o_i^* \wedge o_j^* \wedge o_k^* = 0 \tag{12}$$

for any three indices $i, j, k \in \{1, ..., n\}$. To describe the singular position in CGA completely, we add that for a wheel position Q_i on the link P_i the wheel axle is expressed as

$$o_i^* = (Q_i \wedge e_\infty) \cdot (P_i \wedge e_\infty).$$

6. Conclusions

We proposed a novel analytic description of a snake-like robot, see [11], model by means of conformal geometric algebra. More precisely, the formulae for forward kinematics (3), differential kinematics (4), nonholonomic constrains (6) and singular positions (12) were translated. The CGA-based solutions of robotic systems are actually developing topic, see [3], yet its use to snake-like robots has not been studied. Currently, our simulations are run in Maple, package Cliffordlib developed by R. Ablamowicz and B. Fauser with certain optimization elements, see [5]. Let us note that the final realization should be implemented by means of a system like Gaalop, [4], to reduce the computational complexity, for an example with particular data and hardware solution see [6]. Furthermore in the CGA setting, the mechanism modifications are quite straightforward, particularly the link lengths and the wheel positions may be modified with minimal equation change.

References

[1] Chirikjian, G.S., Burdick, J.W.: An obstacle avoidance algorithm for hyper-redundant manipulators. In: Proceedings 1990 IEEE International Conference on Robotics and Automation, pp. 625–631 (1990)

[2] Dowling, K.J.: Limbless Locomotion: learning to Crawl with a Snake Robot. PhD thesis, Carnegie Melon University, Pittsburgh, USA (1997)

[3] Gonzalez-Jimenez, L., Carbajal-Espinosa, O., Loukianov, A., Bayro-Corrochano, E.: Robust Pose Control of Robot Manipulators Using Conformal Geometric Algebra. Adv. Appl. Clifford Algebr. **24**(2), 533–552 (2014)

[4] Hildenbrand, D.: Foundations of Geometric Algebra Computing. Springer, New York (2013)

[5] Hildenbrand, D., Fontijne, D., Wang, Y., Alexa, M., Dorst, L.: Competitive runtime performance for inverse kinematics algorithms using conformal geometric algebra. In: Proceedings of EUROGRAPHICS 2006 (2006)

[6] Hildenbrand, D., Lange, H., Stock, F., Koch, A.: Efficient inverse kinematics algorithm based on conformal geometric algebra Using reconfigurable hardware. In: Proceedings of the 3rd International Conference on Computer Graphics Theory and Applications (2008)

[7] Hirose, S.: Biologically inspired robots (snake-like locomotor and manipulator). Oxford University Press, Oxford (1993)

[8] Hrdina, J., Vašík, P.: Notes on differential kinematics in conformal geometric algebra approach. Mendel Adv. Intell. Syst. Comput. **378**, 363–374 (2015)

[9] Hrdina, J., Návrat, A., Vašík, P.: 3-Link robotic snake control based on CGA. Adv.Appl. Clifford Algebr. (in print), 1–12 online (2015)

[10] Liljebäck, P., Pettersen, K.Y., Stavdahl, Ø., Gravdahl, J.T.: Snake Robots, Modelling, Mechatronics and Control. Springer, New York (2013)

[11] Matoušek, R., Návrat, A.: Trident snake control based on CGA. Mendel Adv. Intell. Syst. Comput. **378**, 375–385 (2015)

[12] Murray, R.M., Zexiang, L., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton (1994)

[13] Ostrowski, J.: The Mechanics of Control of Undulatory Robotic Locomotion. PhD thesis, CIT (1995)

[14] Perwass, C.: Geometric Algebra with Applications in Engineering. Springer, New York (2009)

[15] Selig, J.M.: Geometric Fundamentals of Robotics. Springer, New York (2004)

[16] Zamora-Esquivel, J., Bayro-Corrochano, E.: Kinematics and differential kinematics of binocular robot heads. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, pp. 4130–4135 (2006)

Jaroslav Hrdina, Aleš Návrat and Petr Vašík
Institute of Mathematics, Faculty of Mechanical Engineering
Brno University of Technology
Brno, Czech Republic
e-mail: hrdina@fme.vutbr.cz

Aleš Návrat
e-mail: navrat.a@fme.vutbr.cz

Petr Vašík
e-mail: vasik@fme.vutbr.cz

R. Matoušek
Institute of Automation and Computer Science
Faculty of Mechanical Engineering
Brno University of Technology
Brno, Czech Republic
e-mail: matousek@fme.vutbr.cz

Received: January 25, 2016.
Accepted: May 31, 2016.

Appendix 5

[24] Hrdina, J., Návrat, A., Vašík, P., Matoušek, R.: *Geometric control of the trident snake robot based on CGA,*, Adv.Appl. Clifford Algebras, (in print), 1–13 online, 2016.

Adv. Appl. Clifford Algebras

DOI 10.1007/s00006-016-0693-7

Advances in
Applied Clifford Algebras

Geometric Control of the Trident Snake Robot Based on CGA

J. Hrdina, A. Návrat, P. Vašík* and R. Matoušek

Abstract. We demonstrate the theory on the 1-link trident snake and the functionality in the CLUCalc software designed for the computations in Clifford algebra. Local control of a general trident snake robot is solved by means of conformal geometric algebra. It is shown that the model modifications are much easier to handle in this setting. Within this paper, we present an alternative model description only, while all its kinematic properties remain. The equations of the direct and differential kinematics, the Pfaff constraints, the inverse kinematics and the singular postures are discussed and translated into the language of CGA.

Mathematics Subject Classification. Primary 93C10; Secondary 15A66.

Keywords. Bionics, Conformal geometric algebra, Clifford algebra, Mathematical robotics, Nonholonomic mechanics, Local controllability.

1. Introduction

Originally the general trident snake robot has been introduced in [8]. It is a planar robot with a body in the shape of a triangle and with three legs consisting of ℓ links. Its precise description is given below. Then, its simplest non-trivial version, corresponding to $\ell = 1$, has been mainly discussed, see e.g. [9,10]. Within this paper, we focus on the general case of ℓ-links. The aim of this article is to solve the complete local control in a new geometric form.

In terms of generalized coordinates, the non-holonomic forward kinematics equations can be understood as a Pfaff system and its solution as a distribution in the configuration space. Rachevsky–Chow theorem implies

The research was supported by the project NETME CENTRE PLUS (LO1202). The results of the project NETME CENTRE PLUS (LO1202) were co-funded by the Ministry of Education, Youth and Sports within the support programme "National Sustainability Programme I".
*Corresponding author.

Published online: 11 June 2016

that the appropriate non-holonomic system is locally controllable if the corresponding distribution is not integrable and the span of the Lie algebra generated by the controlling distribution has to be of the same dimension as the configuration space. The spanned Lie algebra is then naturally endowed by a filtration which shows the way to realize the motions by means of the vector field brackets [13,15]. In the case $\ell = 1$, the system is locally controllable and the filtration is $(3, 6)$.

The classical approach composes the kinematic chain of homogeneous matrices using the moving frame methods and Euler angles, [11]. Instead of this, our aim is to use the notions of conformal geometric algebra (CGA), where the Euclidean space $\mathbb{E}_3$ is included. In this geometric setting, we can easily handle both linear objects and spheres of dimensions 2, 1 and 0, see [1,2,14,16].

In particular, the 0-dimensional sphere, referred to as a point pair, is used to derive the kinematic equations and for the control of the non-holonomic snake like robotic mechanisms, consequently. More precisely, to any link of a single point pair is assigned and the mechanism is transformed by rotations and translations. We introduce the forward kinematic Eq. (6), the differential kinematic Eq. (7), as well as the non-holonomic constraints (8). We also derive an equation for singular positions of the robot (10).

We demonstrate the theory on the 1-link trident snake and the functionality in the CLUCalc software designed for the computations in Clifford algebra, particularly in conformal geometric algebra.

The advantage of CGA description lies in the simplicity of the model modifications, i.e. variable link length and variable wheel position. Furthermore, we fully use the advantage of CGA in operations representation, precisely rotations and translations are represented by particular CGA elements. Note that the solutions of local controllability, motion control model and singular points analysis can be adopted from the classical models, see [8]. Within this paper, we present an alternative model description only, while all its kinematic properties remain.

2. Conformal Geometric Algebra: CGA

The classical approach composes the kinematic chain of homogeneous matrices using the moving frame methods and Euler angles, or, in a more advanced way, the quaternion algebra $\mathbb{H}$ by conjugation $x \mapsto q^{-1}xq$, where we view an Euclidean point x as a quaternion

$$x = (x_1, x_2, x_3) \rightleftharpoons x_1 i + x_2 j + x_3 k.$$

and q is a quaternion given by

$$q = \cos\tfrac{\theta}{2} + u \sin\tfrac{\theta}{2},$$

where u is an axis of rotation $u_1 i + u_2 j + u_3 k$. Instead of this, we use the notions of conformal geometric algebra, i.e. the Clifford algebra $\mathcal{C}l(4, 1)$ where the Euclidean space $\mathbb{E}_3$ is included by a mapping $x \mapsto x + \frac{1}{2}x^2 e_\infty + e_0$. Consequently, we can easily handle both linear objects and spheres of any

dimension. Namely, these objects are simply elements of the algebra and can be transformed and intersected with ease. In addition, rotations, translation, dilations and inversions all become rotations in our 5-dimensional space, see [2,14,16].

More precisely, let $\mathbb{R}^{4,1}$ denote a vector space $\mathbb{R}^5$ equipped with the scalar product of signature $(4,1)$ and let $\{e_1, e_2, e_3, e_+, e_-\}$ be an orthonormal basis. The Clifford algebra $\mathcal{C}l(4,1)$ can be described as a free, associative and distributive algebra such that the *geometric product* e_ie_j

(i) coincides with the scalar product in the case $i = j$,
(ii) is equal to $-e_je_i$ for $i \neq j$. Hence the dimension of the algebra is $2^5 = 32$.

Next to the geometric product, we define two additional products on $\mathbb{R}^{4,1}$ based on the geometric one for any $u, v, \in \mathbb{R}^{4,1}$, *inner product* and *wedge product*, respectively:

$$u \cdot v = \frac{1}{2}(uv + vu), \quad u \wedge v = \frac{1}{2}(uv - vu)$$

and thus for the basis elements we have $uv = u \cdot v + u \wedge v$. Note that $u \cdot v$ is a scalar (an algebra element of grade zero) while $u \wedge v$ is a bivector (an algebra element of grade two). In this sense, the definition of these product extends to the whole algebra. Namely, given two basis blades $E_k = e_{a_1} \wedge \cdots \wedge e_{a_k}$ and $E_l = e_{a_1} \wedge \cdots \wedge e_{a_l}$ of grades k and l, respectively, the wedge (outer) product is defined as

$$E_k \wedge E_l := \langle E_k E_l \rangle_{k+l}$$

while the inner product is defined as

$$E_k \cdot E_l := \langle E_k E_l \rangle_{|k-l|},$$

where $\langle \ \rangle_k$ is the grade projection into the grade k. These products can be used effectively to compute an intersection of geometric objects and distances, respectively.

The conformal geometric basis elements can be represented by the multi-vectors from $\mathcal{C}l(4,1)$ either in the outer product null space (OPNS) representation or in its dual, so called inner product null space (IPNS) representation. To work with CGA effectively, one defines $e_0 = \frac{1}{2}(e_- + e_+)$ and $e_\infty = (e_- - e_+)$. Consequently, the following properties hold. The geometric objects which we use in this paper are then given as follows.

Object	CGA element (OPNS)
Point	$Q = x + \frac{1}{2}x^2 e_\infty + e_0$
Point pair Q_1, Q_2	$P = Q_1 \wedge Q_2$
Line L	$L = Q_1 \wedge Q_2 \wedge e_\infty$

Each geometric transformation (rotation, translation, dilation, inversion) of a geometric object represented by an algebra element $\mathcal{O}$ is realized by conjugation $\mathcal{O} \mapsto M\mathcal{O}\tilde{M}$, where M is an appropriate multi-vector and $\tilde{M}$ is its reversion. For instance, the translation in the direction $t = t_1e_1 + t_2e_2 + t_3e_3$ is realized by conjugation by the multi-vector

$$T = 1 - \tfrac{1}{2}te_\infty,$$

which can be written as $e^{-\frac{1}{2}te_\infty}$, and the rotation around the axis L by angle ϕ is realized as conjugation by the multi-vector

$$R = \cos\tfrac{\phi}{2} - L\sin\tfrac{\phi}{2}.$$

Similarly to the case of a translation, the rotation can be also written as $\mathrm{e}^{-\frac{1}{2}\phi L}$.

3. Control Theory of the Trident Snake Robot

The model of a general trident snake robot is illustrated in Fig. 1. It is a planar robot which consists of a body in the shape of an equilateral triangle with circumscribed circle of the unit radius and three branch legs. Each of the legs consists further of ℓ rigid links of constant unit length interconnected by motorised joints and linked with the vertices of the triangular body by motorised joints. Each link has a passive wheel at its centre which provide an important snake-like property that the ground friction in the direction perpendicular to the link is considerably higher than the friction of a simple forward move. In particular, this prevents the slipping and sliding sideways. We assume the wheels are placed in the link centres, but the case of a general position on a link of an arbitrary length is also discussed.

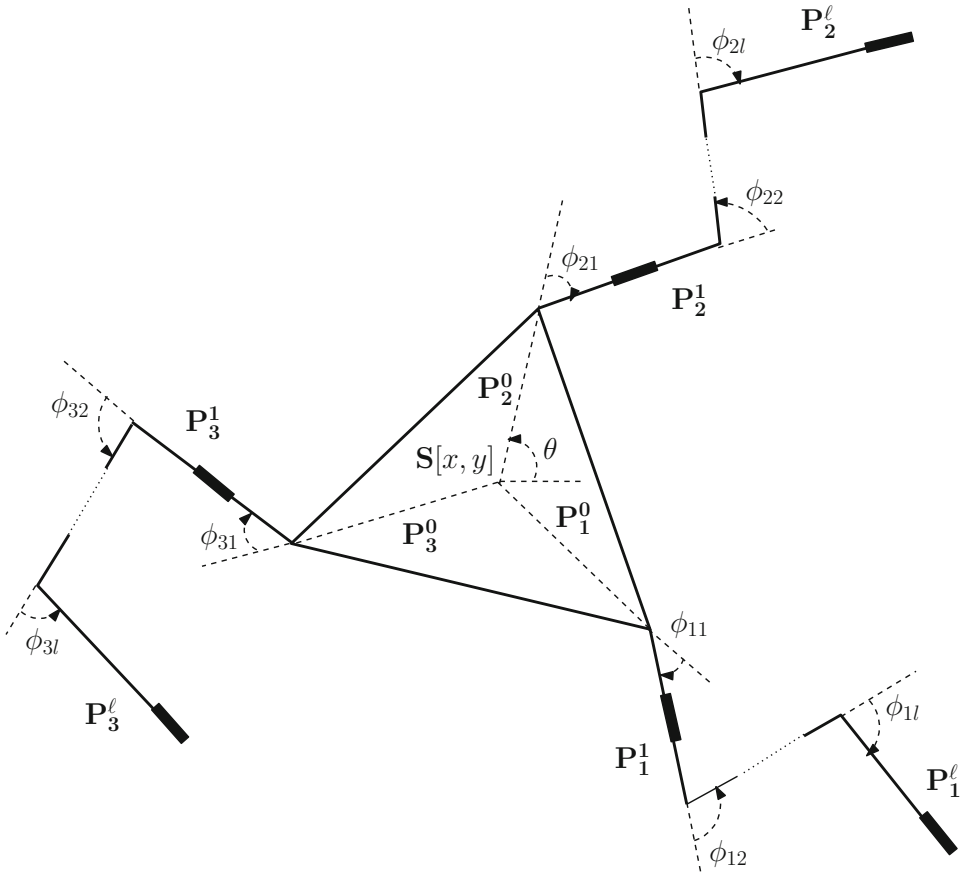


FIGURE 1. Trident snake robot model

To describe the actual position of a trident snake robot we need the set of $3+3l$ generalized coordinates as shown in Fig. 1. According to [8], we call $g := (x, y, \theta) \in \mathbb{SE}(2)$ the configuration vector and

$$\phi := (\phi_{11}, \ldots, \phi_{1\ell}, \phi_{21}, \ldots, \phi_{2\ell}, \phi_{31}, \ldots, \phi_{3\ell}) \in (S^1)^{3\ell}$$

the shape vector, where Φ_{ij} stands for the angle of the j-th link attached to the i-th vertex. Note that we consider the same number of links attached to each vertex but, generally, no restrictions on number of links or other parameters are necessary. Then the generalized coordinates associated to our system are

$$q = (g, \phi) \in \mathbb{SE}(2) \times (S^1)^{3\ell} \tag{1}$$

3.1. Euclidean Description

In this section, we rephrase the results of [8], where the kinematics of the trident snake robot is derived classically using Euclidean geometry. First the Euclidean position vector of link centres is expressed in terms of generalized coordinates (joint angles) via so called moving frame algorithm and then 3ℓ nonholonomic constraints are assembled. This gives the following kinematic equation:

$$A(\phi) R_\phi^T \dot{g} = B(\phi)\dot{\phi},$$

where the matrices occurring in this equation are defined by

$$A(\phi) = \begin{pmatrix} A_1(\phi) \\ A_2(\phi) \\ A_3(\phi) \end{pmatrix} \in \mathbb{R}^{3\ell \times 3}, \; B = \begin{pmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_3 \end{pmatrix} \in \mathbb{R}^{3\ell \times 3\ell},$$

$$A_i = \begin{pmatrix} \sin(\phi_{i1} + \alpha_i) & -\cos(\phi_{i1} + \alpha_i) & -1 - \cos\phi_{i1} \\ \sin(\phi_{i1} + \phi_{i2} + \alpha_i) & -\cos(\phi_{i1} + \phi_{i2} + \alpha_i) & -1 - \cos(\phi_{i1} + \phi_{i2}) - \cos(\phi_{i2}) \\ \vdots & \vdots & \vdots \\ \sin(\beta^i_{j0} + \alpha_i) & -\cos(\beta^i_{j0} + \alpha_i) & -1 - \sum_{k=0}^{l-1} \cos\beta^l_{jk} \end{pmatrix},$$

$$B_i = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ \cos\beta^i_{21} & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \cos\beta^i_{l1} & \cos\beta^i_{l2} & \cdots & 1 \end{pmatrix}$$

and where $\alpha_1 = -\frac{2}{3}\pi$, $\alpha_2 = 0$, $\alpha_3 = \frac{2}{3}\pi$ and $\beta^i_{jk} := \sum_{h=k+1}^{j} \phi_{ih}$ and R_θ is the rotation matrix by the angle θ in the xy-plane. The control inputs are the angular velocities of the joints, i.e. $u := \dot{\phi} \in \mathbb{R}^{3\ell}$. In order to derive the state equation, one defines an input transformation $u = B(\phi)^{-1}A(\phi)v$, where v is a (three dimensional) virtual input. Note that it is always possible since $B(\phi)$ is regular for all ϕ. Then one obtains a control (state) equation $\dot{q} = Gv$, where the control matrix is equal to

$$G = \begin{pmatrix} R_\theta \\ B(\phi)^{-1}A(\phi) \end{pmatrix}. \tag{2}$$

3.2. Direct and Inverse Kinematics via CGA

Now we show how to describe the system exclusively in terms of the conformal geometric algebra. We can view the robot as three ordinary $(\ell+1)$-link snakes joint by their tails such that the tail links has a specific configuration. The CGA description of a simple snake robot motion can be found in [1,7]. Following this idea we denote by Q_k^0 the centre S of the body, and by $Q_k^1, \dots, Q_k^\ell$ the successive joints of the k-th branch leg, $k = 1, 2, 3$. As a central object that describes the state of the system we choose the set of point pairs which represent individual leg links

$$P = (P_1^0, \dots, P_1^\ell, P_2^0, \dots, P_2^\ell, P_3^0, \dots, P_3^\ell). \tag{3}$$

These point pairs are computed in terms of the wedge product in CGA as $P_k^i := Q_k^i \wedge Q_k^{i+1}$. On the other hand, Q_k^i is easily extracted from the point pair by a projection

$$\frac{-\sqrt{P_k^i \cdot P_k^i} + P_k^i}{-e_\infty \cdot P_k^i}.$$

Consequently, we may freely switch between point pairs and points defining their ends. Of course, not all triples of pair points define a state of the robot. In terms of the CGA inner product, the consistency relations read $Q_j^1 \cdot Q_k^1 = -\frac{3}{2}$ for each $j \neq k$, and $Q_k^i \cdot Q_k^{i+1} = -\frac{1}{2}$ for each $i = 0, \dots, \ell$. These equations read that the tail points have the constant distance $\sqrt{3}$ and the length of links is 1, respectively. Implicitly, it also says that Q_k^0 are equal.

Having such an admissible state (3), we can assess the kinematic equations. At first, let us look at the zero position $q = 0$. Since $Q_2^i(0) = [i, 0]$, the elements in CGA corresponding to P_2^i are established as

$$\begin{aligned} P_2^i(0) &= \left(ie_1 + \tfrac{1}{2}i^2 e_\infty + e_0\right) \wedge \left((i+1)e_1 + \tfrac{1}{2}(i+1)^2 e_\infty + e_0\right) \\ &= \tfrac{1}{2}i(i+1)e_{1\infty} - e_{10} - \tfrac{1}{2}(2i+1)e_{\infty 0}, \end{aligned} \tag{4}$$

where we have used a shortened notation $e_{1\infty} = e_1 \wedge e_\infty$ etc. The algebra elements $P_{1,3}^i(0)$ which correspond to the zero position of the links of the first and the third branch leg are obtained from the corresponding links of the first branch by rotation by angle $\frac{2}{3}\pi$ and $-\frac{2}{3}\pi$ respectively, i.e.

$$P_{1,3}^i(0) = (\tfrac{1}{2} \pm \tfrac{\sqrt{3}}{2}e_{12}) P_2^i(0) (\tfrac{1}{2} \mp \tfrac{\sqrt{3}}{2}e_{12}). \tag{5}$$

The particular point pairs in a general position q as in (1) are obtained by a translation to $[x, y]$ composed by a trident body rotation θ and a series of rotations of the corresponding leg links by angles ϕ_{ki}. In CGA, it is expressed for each $k = 1, 2, 3$ and $i = 0, \dots, \ell$ as a conjugation

$$\begin{aligned} P_k^i(q) &= M_k^i(q) P_k^i(0) \widetilde{M}_k^i(q), \\ M_k^i(q) &= R_{\phi_{ki}} \cdots R_{\phi_{k1}} R_\theta T_{x,y}, \end{aligned} \tag{6}$$

where the translation $T_{x,y}$ and the rotations $R_{\phi_{ki}}$ are given by

$$\begin{aligned} T_{x,y} &= 1 - \tfrac{1}{2}(xe_1 + ye_2)e_\infty, \\ R_{\phi_{ki}} &= \cos\tfrac{\phi_{ki}}{2} - L_{ki} \sin\tfrac{\phi_{ki}}{2}, \end{aligned}$$

and where the axes of rotations are given by

$$L_{ki} = (1 - \tfrac{1}{2}Q_k^i e_\infty)e_{12}(1 + \tfrac{1}{2}Q_k^i e_\infty).$$

Note that we have used the notation $\phi_{k0} = \theta$ for each $k = 1, 2, 3$ and note that the procedure is recursive. Namely, given a point pair P_k^i, we compute the projection Q_k^{i+1} first, then we compute $L_{(i+1)k}$. From this axis we compute $R_{\phi_{k(i+1)}}$ and then by (6) we get P_k^{i+1}.

The CGA approach is convenient also for solving problems of the inverse kinematics. In CGA, it can be done in a geometrically very intuitive way due to its easy handling of intersections of geometric objects like spheres, circles, planes. A basic problem is to find the generalized coordinates in terms of the robot position. In our case, having an admissible state (3), we first compute the centre $S = Q_k^0$ by a projection of a point pair P_k^0, and for each $i = 1, 2, 3$ we form lines through two consecutive links P_k^i and P_k^{i+1} . Then we compute the coordinates via the inner product as

$$\begin{aligned} x &= S \cdot e_1, \\ y &= S \cdot e_2, \\ \cos\theta &= (P_2^0 \wedge e_\infty) \cdot e_{1\infty 0}, \\ \cos\phi_{ki} &= (P_k^{i-1} \wedge e_\infty) \cdot (P_k^i \wedge e_\infty). \end{aligned}$$

3.3. Differential Kinematics and Singular Points via CGA

Let us now determine the velocity from the direct kinematics, which is obtained by differentiating (6). It is proved in [16] and generalized in [6] that the total differential of a general kinematic chain

$$\mathcal{O} = R_1 \dots R_n \mathcal{O}(0) \tilde{R}_n \dots \tilde{R}_1$$

containing any geometric object $\mathcal{O}$ and rotations $R_1, \dots R_n$ is equal to

$$d\mathcal{O} = \sum_{j=1}^{n} (\mathcal{O} \cdot L_j) dq_j,$$

where $\mathcal{O} \cdot L_j$ is the inner product of the geometric object (in the actual position) and the axis of the rotation L_j. This formula follows basically from the fact that each rotation can be expressed as an exponential. But the same is true for translations since we may view each translation as a degenerate rotation, with an 'axis' containing e_∞. Hence the formula above holds also if we allow R_i to be a generalized rotation, i.e. a rotation or a translation.

In our case, the equation of the direct kinematics is given by (6). A nice consequence of the geometric formulation is that the same equation holds for an arbitrary chosen object Q attached to P_k^i, i.e. $Q = M_k^i(q)Q(0)\widetilde{M}_k^i(q)$. The differentiation of this kinematic chain then yields the following differential formula for any Q on P_k^i:

$$\dot{Q} = (Q \cdot e_{1\infty})\dot{x} + (Q \cdot e_{2\infty})\dot{y} + \sum_{j=0}^{i} (Q \cdot L_{kj})\dot{\phi}_{kj}. \tag{7}$$

This equation can be seen as an analogue of the classical equation of differential kinematics. Namely, we have a system in the usual form $\dot{Q} = J\dot{q}$ but the

entries of the 'Jacobi' matrix J are the inner products of a point and an axis and thus they belong to the algebra (and not to a field of functions). If Q is the position of a wheel, then, as the wheels do not slip to the side direction, its velocity is parallel to P_k^i, which in CGA reads

$$\dot{Q} \wedge P_k^i \wedge e_\infty = 0. \tag{8}$$

Now, if we substitute (7) in (8), we obtain a system of linear ODEs, which can be written in a classical form $A\dot{q} = 0$, where the Pfaff matrix A is given by

$$A_{ij} = J_{ij} \wedge P_k^i \wedge e_\infty, \tag{9}$$

provided that if

- $k = 1$ then $j = 1, \ldots, l$,
- $k = 2$ then $j = l + 1, \ldots, 2l$,
- $k = 3$ then $j = 2l + 1, \ldots, 3l$.

In our particular case, the wheels are located at the link centres $Q = P_k^i e_\infty \widetilde{P}_k^i$, for each $i = 1, \ldots, \ell - 1$, and at the head point of each branch $Q = Q_k^\ell$. Hence we get a system of 3ℓ first order differential equations with $3\ell + 3$ variables. It is easy to see that each A_{ij} is a multiple of $(e_3)^*$. Thus the Pfaff equation $A\dot{q} = 0$ can be solved for A considered as a matrix over the field of functions. Then, in a point, one obtains $\dot{q} = Gu$ for a suitable control matrix G and an input u. Substituting this into (7) one obtains a system $\dot{Q} = JGu$. Let us also remark that the particular position of wheels does not play any role formally. The Eqs. (6), (7), (8) and (9) are valid for any position and also for an arbitrary lengths of leg links.

At the end of this section, let us discuss the positions of the robot which are critical for the control. Such critical (singular) positions occur when the velocity constraints degenerate. It is e.g. in the case when all wheel axles are either parallel or intersect in one point, so called instantaneous center of rotation (ICR). Obviously, the former singular positions coincide with the latter if the centre of rotation is in infinity. Thus, in CGA, we have one equation describing such singular positions. Denoting by o_k^i the i-th wheel axle on the k-th branch it reads

$$\bigwedge o_k^{i*} = 0, \tag{10}$$

where o_k^{i*} denotes the IPNS representation of o_k^i. We can equivalently express this equation in terms of the geometric product as $\langle o_1^1 \ldots o_3^l \rangle_1 = 0$. If we denote by Q the position of a wheel attached to the link P_k^i , then its axle o_k^i is computed by the formula

$$o_k^i = (Q \wedge e_\infty) \cdot (P_k^i \wedge e_\infty). \tag{11}$$

4. 1-Link Trident Snake Robot

Here we demonstrate the theory on the simplest non-trivial case, i.e. the case $\ell = 1$. We also assume that the leg links have the unit length and that the wheels are attached at the ends of these links. For more details see [12].

4.1. Kinematics

For simplicity we omit the upper index denoting the first link from now on. By (4) and (5), the zero position of point pairs associated to the leg links is given by

$$\begin{aligned} P_2(0) &= e_{1\infty} - e_{10} - \tfrac{3}{2}e_{\infty 0}, \\ P_{1,3}(0) &= -\tfrac{1}{2}e_{1\infty} + \tfrac{1}{2}e_{10} \pm \tfrac{\sqrt{3}}{2}e_{2\infty} \mp \tfrac{\sqrt{3}}{2}e_{20} - \tfrac{3}{2}e_{\infty 0}. \end{aligned}$$

By (6), the general position of a point pair P_k is given by

$$P_k = R_{\phi_{k1}} R_\theta T_{x,y} P_k(0) \tilde{T}_{x,y} \tilde{R}_\theta \tilde{R}_{\phi_{k1}}.$$

The same kinematic chain holds for any point Q on P_k. Thus, for the leg ends $Q = (Q_1, Q_2, Q_3)^T$, we get a differential equation $\dot{Q} = J\dot{q}$, where

$$J = \begin{pmatrix} Q_1 \cdot e_{1\infty} & Q_1 \cdot e_{2\infty} & Q_1 \cdot L_0 & Q_1 \cdot L_1 & 0 & 0 \\ Q_2 \cdot e_{1\infty} & Q_2 \cdot e_{2\infty} & Q_2 \cdot L_0 & 0 & Q_2 \cdot L_2 & 0 \\ Q_3 \cdot e_{1\infty} & Q_3 \cdot e_{2\infty} & Q_3 \cdot L_0 & 0 & 0 & Q_3 \cdot L_3 \end{pmatrix}.$$

The nonholonomic constraints (8) read $\dot{Q}_i \wedge P_i \wedge e_\infty = 0$. The fact that any rotation R_i commutes with its axis of rotation L_i imply that $(Q_i \cdot L_i) \wedge P_i \wedge e_\infty$ is constant and equal to the pseudoscalar for each $i = 1, 2, 3$. Hence the Pfaff matrix (9) has the block form $A = (Q_i \cdot \ell_j \wedge P_i \wedge e_\infty | 1)$, where ℓ_j denotes $e_{1\infty}$, $e_{2\infty}$ and L_0, respectively, and the control matrix G is a 6×3 matrix

$$G = \begin{pmatrix} 1 \\ -Q_i \cdot \ell_j \wedge P_i \wedge e_\infty \end{pmatrix}.$$

Evaluating the CGA products we obtain that the distribution G is spanned by vector fields

$$\begin{aligned} g_1 &= \partial_x + \sin(\theta + \phi_1)\partial_{\phi_1} + \sin\left(\theta + \phi_2 + \tfrac{2\pi}{3}\right)\partial_{\phi_2} + \sin\left(\theta + \phi_3 + \tfrac{4\pi}{3}\right)\partial_{\phi_3}, \\ g_2 &= \partial_y - \cos(\theta + \phi_1)\partial_{\phi_1} - \cos\left(\theta + \phi_2 + \tfrac{2\pi}{3}\right)\partial_{\phi_2} - \cos\left(\theta + \phi_3 + \tfrac{4\pi}{3}\right)\partial_{\phi_3}, \\ g_3 &= \partial_\theta - (1 + \cos\phi_1)\partial_{\phi_1} - (1 + \cos\phi_2)\partial_{\phi_2} - (1 + \cos\phi_3)\partial_{\phi_3}. \end{aligned}$$

It is easy to check that it corresponds to the general result (2) obtained by the Euclidean geometry. By a direct computation, one can also check that the equation (10) describing the singular points gives exactly the equation in [8, Remark 1].

By [8], these vector fields define a bracket generating distribution with growth vector $(3, 6)$ in the regular points. It means that in each regular point the vectors g_1, g_2, g_3 together with their Lie brackets span the whole tangent space. Consequently, the system is controllable by Chow–Rashevsky theorem.

4.2. A Modification

Now we demonstrate the efficiency of our geometric approach. Let us allow the trident robot to have general lengths l_1, l_2, l_3 of legs. The only thing that differ are formulas (4) and (5) for the zero position of the robot. Now we have

$$P_2(0) = \tfrac{1}{2}l_2(l_2+1)e_{1\infty} - l_2e_{10} + \tfrac{1}{2}l_2(l_2+2)e_{0\infty},$$
$$P_{1,3}(0) = -\tfrac{1}{4}l_{1,3}(l_{1,3}+1)e_{1\infty} \pm \tfrac{\sqrt{3}}{4}l_{1,3}(l_{1,3}+1)e_{2\infty},$$
$$+ \tfrac{1}{2}l_{1,3}e_{10} \mp \tfrac{\sqrt{3}}{2}l_{1,3}e_{20} + \tfrac{1}{2}l_{1,3}(l_{1,3}+2)e_{0\infty}.$$

All other formulas are universal and thus remain unchanged. The control fields of the modified trident snake robot are then

$$g_1 = \partial_x + \tfrac{1}{l_1}\sin(\theta+\phi_1)\partial_{\phi_1} + \tfrac{1}{l_2}\sin\left(\theta+\phi_2+\tfrac{2\pi}{3}\right)\partial_{\phi_2} + \tfrac{1}{l_3}\sin\left(\theta+\phi_3+\tfrac{4\pi}{3}\right)\partial_{\phi_3},$$
$$g_2 = \partial_y - \tfrac{1}{l_1}\cos(\theta+\phi_1)\partial_{\phi_1} - \tfrac{1}{l_2}\cos\left(\theta+\phi_2+\tfrac{2\pi}{3}\right)\partial_{\phi_2} - \tfrac{1}{l_3}\cos\left(\theta+\phi_3+\tfrac{4\pi}{3}\right)\partial_{\phi_3},$$
$$g_3 = \partial_\theta - \tfrac{1}{l_1}(l_1+\cos\phi_1)\partial_{\phi_1} - \tfrac{1}{l_2}(l_2+\cos\phi_2)\partial_{\phi_2} - \tfrac{1}{l_3}(l_3+\cos\phi_3)\partial_{\phi_3}$$

and the singular position corresponds to the angles that satisfy

$$(l_1+\cos\phi_1)\cos(\phi_3-\phi_2+\tfrac{\pi}{6}) + (l_2+\cos\phi_2)\cos(\phi_1-\phi_3+\tfrac{\pi}{6}),$$
$$+(l_3+\cos\phi_3)\cos(\phi_1-\phi_2+\tfrac{\pi}{6}) = 0.$$

4.3. CLUCalc Implementation

The proposed trident snake control was tested in CLUCalc software [2,14], which is designed exactly for calculations in arbitrary predefined geometric algebra. The following code piece contains the definition of the initial position:

```
// INITIAL POSITION\\
S0=VecN3(0,0,0);\\
LB0=VecN3(0,0,1);\\
R=RotorN3(0,0,1,2*Pi/3);\\
// Joints\\
pb10=VecN3(1,0,0);\\
pb20=R*pb10*\textasciitilde R;\\
pb30=R*pb20*\textasciitilde R;\\
// Axes\\
L10=TranslatorN3(pb10)*LB0*TranslatorN3(-pb10);\\
L20=R*L10*\textasciitilde R;\\
L30=R*L20*\textasciitilde R;\\
// Ends of legs\\
p10=VecN3(2,0,0);\\
p20=R*p10*\textasciitilde R;\\
p30=R*p20*\textasciitilde R;\\ \\
```

The initial position is then recalculated with respect to the controlling parameters change to get the current position. The code we demonstrate corre-

FIGURE 2. g_1 Direction (pictured by CLUCalc)

FIGURE 3. g_2 Direction (pictured by CLUCalc)

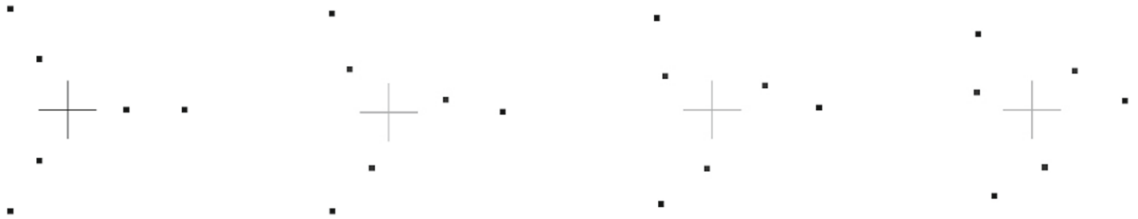


FIGURE 4. g_3 Direction (pictured by CLUCalc)

sponds to the body and the first leg of the trident snake robot. The other legs are computed in the same way.

```
T=TranslatorN3(x,y,0);\\
// BODY\\
 // Center\\
 S=T*S0*\textasciitilde T;\\
 // Axis \\
 LB=T*LB0*\textasciitilde T;\\
 // Motor\\
 MB=TranslatorN3(LB)*RotorN3(0,0,1,d)*\textasciitilde TranslatorN3(LB); \\
\\
// FIRST LEG \\
// Joint\\
 :Blue; \\
 :pb1=MB*T*pb10*\textasciitilde T*\textasciitilde MB;\\
 // Axis\\
 L1=MB*T*L10*\textasciitilde T*\textasciitilde MB;\\
 // Motor\\
 M1=TranslatorN3(L1)*RotorN3(0,0,1,a)*TranslatorN3(-L1);    \\
 // End\\
  :Black;\\
 :p1=M1*MB*T*p10*\textasciitilde T*\textasciitilde MB*\textasciitilde M1; \\
```

Figures 2, 3, 4 demonstrate the evolution from 0 in the direction of g_1, g_2 and g_3 vector fields, respectively.

Note that the inner triple of points defines the root vertices and the outer three points show the wheel positions. The cross denotes the origin.

5. Conclusions

The snake-like robot modelling by means of CGA is a quite promising topic. This approach leads to a list of nice formulas for forward and inverse kinematics, differential kinematics, nonholonomic constrains and singular positions. It brings not only a geometric point of view to the topic that is important for the complete problem understanding, but it also allows the calculations to be done faster if a system like Gaalop is used [2]. Yet, our current simulations are run in Maple, package Cliffordlib developed by Ablamowicz and Fauser which contains certain optimization elements, see [3]. The computational complexity analysis for a point pair inverse kinematics optimized algorithms w.r.t. particular hardware solution can be found in [4].

To demonstrate the contribution of CGA calculus let us restrict from CGA to quaternions. Classically, rotations in 3D-space are realized by multiplication of a 3×3 matrix. The composition of rotations is then rotation matrices multiplication, i.e such composition is given by 27 multiplications and 18 summations. If you represent rotations by quaternions, for the same composition you need just 16 multiplications and 15 summations. This reduces the computational time remarkably.

References

[1] Gonzalez-Jimenez, L., Carbajal-Espinosa, O., Loukianov, A., Bayro-Corrochano, E.: Robust pose control of robot manipulators using conformal geometric algebra. Adv. Appl. Clifford Algebr. **24**(2), 533–552 (2014)

[2] Hildenbrand, D.: Foundations of Geometric Algebra Computing. Springer, New York (2013)

[3] Hildenbrand, D., Fontijne, D., Wang, Y., Alexa, M., Dorst, L.: Competitive runtime performance for inverse kinematics algorithms using conformal geometric algebra. In: Proceedings of EUROGRAPHICS 2006 (2006)

[4] Hildenbrand, D., Lange, H., Stock, F., Koch, A.: Efficient inverse kinematics algorithm based on conformal geometric algebra Using reconfigurable hardware. In: Proceedings of the 3rd International Conference on Computer Graphics Theory and Applications (2008)

[5] Hirose, S.: Biologically inspired robots (snake-like locomotor and manipulator). Oxford University Press, Oxford (1993)

[6] Hrdina, J., Vašík, P.: Notes on differential kinematics in conformal geometric algebra approach. In: Mendel 2015, Advances in Intelligent Systems and Computing, vol. 378, pp. 363–374 (2015)

[7] Hrdina, J., Návrat, A., Vašík, P.: 3-Link robotic snake control based on CGA. Adv. Appl. Clifford Algebr, pp. 1–12 (2015) (in print)

[8] Ishikawa, M: Trident snake robot: locomotion analysis and control. NOLCOS. IFAC Symp. Nonlinear Control Syst. **6**, 1169–1174 (2004)

[9] Ishikawa, M., Minami, Y., Sugie, T.: Development and control experiment of the trident snake robot. In: Proceedings of the 45th IEEE Conference on Decision and Control (CDC2006), pp. 6450–6455 (2006)

[10] Ishikawa, M., Minami, Y., Sugie, T.: Development and control experiment of the trident snake robot. IEEE/ASME Trans. Mechatron. **15**(1), 9–15 (2010)

[11] Liljebäck, P., Pettersen, K.Y., Stavdahl, Ø., Gravdahl, J.T.: Snake Robots, Modelling, Mechatronics and Control. Springer, New York (2013)

[12] Matoušek, R., Návrat, A.: Trident snake control based on CGA. In: Matousek, R. (ed.) Mendel 2015, Advances in Intelligent Systems and Computing, vol. 378, pp. 375–385. Springer, Heidelberg (2015)

[13] Murray, R.M., Zexiang, L., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton, FL (1994)

[14] Perwass, C.: Geometric Algebra with Applications in Engineering. Springer, New York (2009)

[15] Selig, J.M.: Geometric Fundamentals of Robotics. Springer, New York (2004)

[16] Zamora-Esquivel, J., Bayro-Corrochano, E.: Kinematics and differential kinematics of binocular robot heads. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, pp. 4130–4135 (2006)

J. Hrdina, A. Návrat and P. Vašík
Institute of Mathematics
Faculty of Mechanical Engineering
Brno University of Technology
Brno, Czech Republic
e-mail: vasik@fme.vutbr.cz

J. Hrdina
e-mail: hrdina@fme.vutbr.cz

A. Návrat
e-mail: navrat@fme.vutbr.cz

R. Matoušek
Institute of Automation and Computer Science
Faculty of Mechanical Engineering
Brno University of Technology
Brno, Czech Republic
e-mail: matousek@fme.vutbr.cz

Received: January 22, 2016.
Accepted: May 30, 2016.