

## MOTION REPRESENTATIONS FOR THE LAFFERRIERE–SUSSMANN ALGORITHM FOR NILPOTENT CONTROL SYSTEMS

IGNACY DULĘBA, JACEK JAGODZIŃSKI

Institute of Computer Engineering, Control and Robotics  
Wrocław University of Technology, ul. Janiszewskiego 11/17, 50–372 Wrocław, Poland  
e-mail: {ignacy.duleba, jacek.jagodziniski}@pwr.wroc.pl

In this paper, an extension of the Lafferriere–Sussmann algorithm of motion planning for driftless nilpotent control systems is analyzed. It is aimed at making more numerous admissible representations of motion in the algorithm. The representations allow designing a shape of trajectories joining the initial and final configuration of the motion planning task. This feature is especially important in motion planning in a cluttered environment. Some natural functions are introduced to measure the shape of a trajectory in the configuration space and to evaluate trajectories corresponding to different representations of motion.

**Keywords:** control, nilpotent system, algorithm, motion representation.

### 1. Introduction

Many systems of contemporary robotics (mobile platforms and robots, free-floating robots, underactuated manipulators) are subordinated to constraints in the Pfaff form. For most of them, the constraints (coupling velocities and configurations into a single relationship) cannot be integrable. In control practice it is more pleasant to deal with admissible actions rather than inadmissible ones. As a consequence, driftless nonholonomic systems arise from the constraints. Driftless systems have a smaller number of controls than the dimensionality of the configurations space and, therefore, a motion planning task for the systems is difficult even in obstacle free environments.

In the robotic literature, a lot of methods of steering driftless nonholonomic systems were designed (cf. references in the book LaValle (2006)). Methods planning a motion for general driftless systems are usually based on graph-searching techniques, where graphs are prepared by suitably exploring properties of the model and its control space. For very special structures of driftless systems (a unicycle robot, chained form systems, flat systems and their extensions (Rouchon, 2001)), dedicated methods were developed exploiting the structure of the model. However, there is still a need for analytic methods of motion planning that work for a relatively wide class of nonholonomic systems.

One of the most prominent methods of motion plan-

ning for nilpotent nonholonomic systems was introduced by Lafferriere and Sussmann (Lafferriere, 1991; Lafferriere and Sussmann, 1990; 1991) and reported also by Kousoulas and Skiadas (2001) as well as LaValle (2006). Its basic version has been already implemented in the Mathematica package (Koussoulas and Skiadas, 2004). There are two possible ways to extend the Lafferriere–Sussmann (LS) algorithm. The first one is either to transform any nonholonomic system into a nilpotent system and then to use the LS algorithm, or to split a motion planning task into small pieces and apply the LS algorithm for each sub-task compensating a small error due to an unnecessary nilpotent controlled system (Sussmann, 1991). Several authors developed techniques to transform a general system into its nilpotent equivalent (Bellaïche *et al.*, 1993; Struemper, 1998; Vendittelli *et al.*, 2004). However, those approximations are local in nature, very complicated and numerically costly.

Another way to extend the LS algorithm is to modify its components fixed at its original version. In the past, the impact of a reference trajectory, i.e., a curve joining the initial point and final one (Dulęba and Jagodziński, 2008a), on the performance of the LS algorithm was investigated. It was shown that the reference trajectory has no influence on the trajectory performed by the controlled system. The original Campbell–Baker–Hausdorff–Dynkin (CBHD) formula generating piecewise-constant controls was replaced by its generalized version (gCBHD)

(Strichartz, 1987) to extend the class of admissible controls. Usually, to generate controls with the gCBHD formula, a Ph. Hall basis of a free Lie algebra is used. In our previous works, special attention was paid to vary the representation of motion for the LS algorithm because the original LS algorithm admitted only two representations called the backward and forward ones. It was shown how to algorithmically determine the Chen–Fliess–Sussmann (CFS) equation for each representation (Dułęba and Jagodziński, 2008b), and how many representations can be generated (Dułęba, 2009).

In this paper, all results on motion representations will be gathered and extended by checking their impact on the shapes of generated trajectories. The paper is organized as follows. In Section 2 a model of the controlled system and some preliminary Lie algebraic terms are introduced and defined. In Section 3, consecutive steps of the LS algorithm are recalled, with special attention paid to selection of motion representation. In Section 4, quality measures to evaluate trajectories are introduced. In Section 5, the impact of representations on the shape of resulting trajectories for the Brockett integrator and the four dimensional chain system is verified via simulations. Section 6 concludes the paper.

## 2. Preliminaries

A driftless nonholonomic system is described by the equation

$$\dot{\mathbf{q}} = \sum_{i=1}^m \mathbf{g}_i(\mathbf{q})u_i, \quad \dim \mathbf{u} = m < n = \dim \mathbf{q}, \quad (1)$$

where  $\mathbf{q} = (q_1, \dots, q_n)^T$  is a configuration,  $u_i$ ,  $i = 1, 2, \dots, m$  are control inputs and  $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_m$  are smooth vector fields called generators of the system (1). A motion planning task is to find controls  $\mathbf{u}(t) = (u_1(t), \dots, u_m(t))^T$  that steer the system from a given initial configuration  $\mathbf{q}_0$  to the final one  $\mathbf{q}_f$ . Sometimes some extra requirements can be imposed on the task (restrictions on the admissibility of configurations, a class of control, the optimality of motion).

With each generator, its formal counterpart (Lie monomial) can be associated. This mapping can be named formalization. In the formal setting, a formal Lie bracket can be defined. This operation assigns two Lie monomials another Lie monomial and satisfies the Jacobi identity and the antisymmetry property (Reutenauer, 1993). To avoid redundancy (due to the properties), a basis is introduced in the space spanned by Lie monomials. The most frequently used basis is due to Ph. Hall (an efficient way to generate the basis was described by Dułęba (1998)). To any Lie monomial, its degree is also assigned. The generators have degree 1, while compound Lie monomials have a degree that is the sum of its components. The formal Lie mono-

mials transforming back (via inverse mapping to formalization) to the vector fields domain define a Lie algebra of vector fields when the formal Lie bracket is substituted with its coordinate dependent form. The Lie bracket assigns any pair of vector fields  $\mathbf{a}, \mathbf{b}$  assigns other vector fields  $[\mathbf{a}, \mathbf{b}] = \frac{\partial \mathbf{b}}{\partial \mathbf{q}} \mathbf{a} - \frac{\partial \mathbf{a}}{\partial \mathbf{q}} \mathbf{b}$ . The Lie algebra of vector fields is a nilpotent one with the order of nilpotency equal to  $k$  if all vector fields with degrees higher than  $k$  vanish. A controllability requirement is quite natural to plan the motion of a nonholonomic system. According to the Chow theorem (Chow, 1939), the system (1) is small time locally controllable if its Lie algebra is of the full rank ( $= n$ ) at each point in the configuration space. In Table 1, dimensionalities of the maximal controllable space are collected for the number of generators and nilpotency degree varied (it was assumed that each element of the Ph. Hall basis is also independent of the other when projected to the space of vector fields).

Table 1. Possible values of Ph. Hall basis elements for the degree of nilpotency and the number of generators varied.

$m$	degree of nilpotency				
	2	3	4	5	6
2	3	5	8	14	23
3	6	14	32	80	196
4	10	30	90	294	964
5	15	55	205	829	3409
6	21	91	406	1960	9695

## 3. LS algorithm

The algorithm was designed to plan a motion for nilpotent (controllable) nonholonomic control system (1) between two given configurations  $\mathbf{q}_0$  and  $\mathbf{q}_f$  in a collision free environment. The algorithm is composed of three major steps.

**3.1. Calculation of extended controls.** First, a smooth reference trajectory  $\lambda(t) = (\lambda_1(t), \dots, \lambda_n(t))^T$  is selected joining the initial point  $\mathbf{q}_0 = \lambda(0)$  with the final one  $\mathbf{q}_f = \lambda(T)$ , where  $T$  denotes the prescribed time horizon. After generating the Ph. Hall basis based on generators of the system (1) and projecting it onto the vector fields domain, an extended system is formulated,

$$\dot{\mathbf{q}} = \sum_{i=1}^r \mathbf{g}_i(\mathbf{q})v_i, \quad (2)$$

where  $v_i$ ,  $i = 1, 2, \dots, m$  are (virtual) extended control inputs,  $\mathbf{g}_{\text{gen}}$  is a set of generators,  $\mathbf{g}_{\text{add}} = \{\mathbf{g}_{m+1}, \dots, \mathbf{g}_r\}$  are all non-zero vector fields with degrees higher than 1. Because it was assumed that the system is controllable (it satisfies the Lie algebra rank condition), for every  $\mathbf{q}$  we have  $\text{rank}(\mathbf{G})(\mathbf{q}) = n$  with  $\mathbf{G} = (\mathbf{g}_{\text{gen}}, \mathbf{g}_{\text{add}})$ .

Then, the reference trajectory  $\lambda(t)$  is substituted for  $\mathbf{q}$  to compute extended controls,

$$\mathbf{v}(t) = \mathbf{G}^\#(\lambda(t)) \frac{d\lambda}{dt}(t), \quad \forall t \in [0, T] \quad (3)$$

using the Moore–Penrose generalized matrix inverse  $\mathbf{G}^\# = \mathbf{G}^T (\mathbf{G}\mathbf{G}^T)^{-1}$ .

### 3.2. Chen–Fliess–Sussmann equation and its solution.

The extended system (2) has its formal counterpart described by the equation

$$\dot{\mathbf{S}}(t) = \mathbf{S}(t)(v_1\mathbf{B}_1 + v_2\mathbf{B}_2 + \dots + v_{r-1}\mathbf{B}_{r-1} + v_r\mathbf{B}_r), \quad (4)$$

where  $\mathbf{B}_i$  are Ph. Hall basis elements (formal counterparts of vector fields from Eqn. (2)), and  $\mathbf{S}(t)$  is a designed motion operator initialized with  $\mathbf{S}(0) = \mathbf{I}$  ( $\mathbf{I}$  denotes the identity operator). Lafferriere and Sussmann considered in their original works (Lafferriere and Sussmann, 1991) only two possible representations: the backward one,

$$\mathbf{S}(t)(\mathbf{G}) = e^{h_r(t)\mathbf{B}_r} e^{h_{r-1}(t)\mathbf{B}_{r-1}} \dots e^{h_2(t)\mathbf{B}_2} e^{h_1(t)\mathbf{B}_1}, \quad (5)$$

and the forward one,

$$\mathbf{S}(t)(\mathbf{G}) = e^{h_1(t)\mathbf{B}_1} e^{h_2(t)\mathbf{B}_2} \dots e^{h_{r-1}(t)\mathbf{B}_{r-1}} e^{h_r(t)\mathbf{B}_r}, \quad (6)$$

where  $h_i(t)$ ,  $i = 1, \dots, r$  are functions to be determined. The exponential mapping  $e : \mathbf{B} \rightarrow e^{\mathbf{B}}$  used in Eqns. (5) and (6) sends an element of a Lie algebra into a Lie group. The mapping (interpreted in terms of vector fields) has a clear physical interpretation: given a velocity  $\mathbf{B}$  (an element of the Lie algebra) and a given state  $\mathbf{q}$ , it is possible to determine its future state  $e^{\mathbf{B}}(\mathbf{q})$  (an element of a Lie group). Here  $e^{a(t)\mathbf{B}}$  means that  $\mathbf{B}$  is scaled with the function  $a(t)$ , depending on controls.

Let us generalize the motion representations (5) and (6). A minimal requirement to preserve full motion abilities is to have the number of functions  $\mathbf{h}$  describing the motion equal to  $r$  at least. On the other hand, to avoid redundancy (and reduce computational complexity while working with a redundant representation), this number must be equal to  $r$  exactly. Consequently, consider a quite general family of representations,

$$\mathbf{S}(t) = \prod_{i=1}^{\bar{r}} e^{\langle \rho_i, \mathbf{h}(t) \cdot \mathbf{B} \rangle}, \quad (7)$$

where  $\mathbf{h}(t) = (h_1(t), \dots, h_r(t))^T$ ,  $\mathbf{B} = (\mathbf{B}_1, \dots, \mathbf{B}_r)^T$  are Ph. Hall basis elements,  $\langle \cdot, \cdot \rangle$  is the inner product,  $\bar{r}$  ( $1 \leq \bar{r} \geq r$ ) describes how many elementary actions  $e^{(\cdot)}$  are used to generate  $\mathbf{S}(t)$ ,  $\rho_i = (\rho_{i1}, \dots, \rho_{ir})^T$  is the  $i$ -th selection vector, where  $\rho_j \in \{0, 1\}$ , which switches on (1), or off (0) components of the vector  $\mathbf{B}$ . The operation

$\star : \mathbb{R}^n \star \mathbb{R}^n \rightarrow \mathbb{R}^n$  is coordinate-wise argument multiplication. Additionally, each representation described by Eqn. (7) has to satisfy the following condition:

$$\sum_{i=1}^{\bar{r}} \rho_i = (1, \dots, 1)^T. \quad (8)$$

The condition (8) forces that any element of the basis  $\mathbf{B}$  occurs only once in the representation. Because as a basis for a free Lie algebra the Ph. Hall basis was selected, the family of representations (7) and (8) will be called Ph. Hall representations. The concatenation of exponents in Eqns. (5)–(7) should be read from left to right. For example,  $\mathbf{S}(t) = e^{h_1\mathbf{X}+h_2\mathbf{Y}} e^{h_3[\mathbf{X},\mathbf{Y}]}$  means that the state is moved by the action of  $e^{h_1\mathbf{X}+h_2\mathbf{Y}}$  and as a result, acts  $e^{h_3[\mathbf{X},\mathbf{Y}]}$ . In general, actions of exponents do not commute. Selection vectors for this example are  $\rho_1 = (1, 1, 0)^T$ ,  $\rho_2 = (0, 0, 1)^T$  ( $r = 3$ ,  $\bar{r} = 2$ ). There exists one special representation, called canonical, which has one selection vector composed of ones only ( $\bar{r} = 1$ ). It is not difficult

Table 2. Number of  $F(r)$  for  $r$  varied.

$r$	1	2	3	4	5	6	7
$F(r)$	1	3	13	75	541	4683	47293

to determine selection vectors for the forward and backward representation. The vectors are composed of zeroes with a single 1; for the backward representation it lies on the  $(r - i)$ -th coordinate, while for the forward representation—on the  $i$ -th coordinate. Using combinatorial arguments, it was established (Duleba, 2009) that the number of representation, for a given  $r$ , is given by the following formula:

$$F(r) = \sum_{k=1}^r \sum_{i=1}^r (-1)^{k-i} \binom{k}{i} i^r. \quad (9)$$

This number grows rapidly as  $r$  increases, cf. Table 2.

Now, the formal equation (4) with a selected motion representation (7), (8) must be solved. Using the Taylor expansion formula

$$e^{\pm h_k \mathbf{P}_k} = \sum_{i=0}^{\infty} (\pm 1)^i \frac{h_k^i \mathbf{P}_k^i}{i!}, \quad (10)$$

which is finite, due to the nilpotency of the system, each term  $e^{(\cdot)}$  is expanded with Eqn. (10). Then, non-commutative multiplications are performed and coefficients in front of the same terms  $\mathbf{B}_i$  compared. In this way, the Chen–Fliess–Sussmann equation is obtained in the form

$$\dot{\mathbf{h}}(t) = \mathbf{M}(\mathbf{h}(t))\mathbf{v}(t), \quad (11)$$

with the initial condition  $\mathbf{h}(0) = \mathbf{0}$  (resulting from  $\mathbf{S}(0) = \mathbf{I}$ ) and  $\mathbf{v}(t)$  given by Eqn. (3). For a given representation,

the process of deriving the CFS equation can be algorithmized (Dułęba and Jagodziński, 2009), and the calculations are to be performed in off-line mode. It appears that  $M(\mathbf{h})$  is a lower triangular matrix of size  $r \times r$ , with the ones on the main diagonal and all elements of the matrix are polynomials in  $\mathbf{h}$  variables. It is not difficult to solve the differential CFS equation (11), although a numerical integration is required because  $\mathbf{v}(t)$ ,  $t \in [0, T]$  is usually not in an analytical form. Finally, Ph. Hall coefficients  $\mathbf{h}$  at the final time  $\mathbf{h}(T)$  are obtained. The last step of the LS algorithm is to generate with controls assumed as  $\mathbf{S}(t)$  at  $t = T$  using  $\mathbf{h}(T)$ .

**3.3. Generating  $S(T)$  with controls.** In order to produce  $\mathbf{S}(T)$ , its components of the form  $e^{\sum_{i=1}^s h_i(T) \mathbf{B}_i}$  must be generated, where  $h_i(T)$  are some real constants while  $\mathbf{B}_i$  are Lie monomials and  $s \geq 1$ . When  $s = 1$  and  $\mathbf{B}_i$  is just a generator, there is no problem to produce it with controls. The control corresponding to the generator is switched on while all others switched off. To generate either single items or linear combinations including Lie monomials with degrees higher than 1, more sophisticated control scenarios must be used. There are two methods to cope with this case. The first one is to use the Campbell–Baker–Hausdorff–Dynkin (CBHD) formula which says that concatenation of  $e^{\mathbf{A}}$  followed by  $e^{\mathbf{B}}$  (where  $\mathbf{A}, \mathbf{B}$  are generators) equals

$$e^{\mathbf{A}} e^{\mathbf{B}} = e^{\mathbf{A} + \mathbf{B} + \frac{1}{2}[\mathbf{A}, \mathbf{B}] + \frac{1}{12}[\mathbf{A}, [\mathbf{A}, \mathbf{B}]] - \frac{1}{12}[\mathbf{B}, [\mathbf{A}, \mathbf{B}]] + \dots}, \quad (12)$$

i.e., switching on  $\mathbf{A}$  and then  $\mathbf{B}$ , a Lie series is obtained (in fact, the series is finite as we work with nilpotent systems). Applying (12) many times, one can extract only one term from the series annihilating the others. It is worth noticing that this scenario requires piecewise constant controls to switch on and off generators, and to generate Lie monomials with higher degrees many switching actions are performed. For example, to generate  $[\mathbf{A}, \mathbf{B}]$  with the order of nilpotency equal to 2, the scenario  $e^{\mathbf{A}} e^{\mathbf{B}} e^{-\mathbf{A}} e^{-\mathbf{B}}$  can be applied.

The CBHD formula offers a conceptually simple method to generate higher degree Lie monomials with controls. However, piecewise constant controls are difficult to generate with technical means and the number of switching actions can be quite large. More flexibility in generation of vector fields with controls is offered by the generalized version of the CBHD formula (gCBHD) (Strichartz, 1987). The gCBHD formula applied to the system (1) describes locally its solution initialized at a given configuration in the form of Lie series with control-dependent coefficients. The series is finite as nilpotent systems are considered. (A detailed presentation of the gCBHD formula and its impact on motion planning of nonholonomic systems can be found in the work of Dułęba and Khefifi (2006).) To generate the required Lie mo-

nomial or polynomial, a class of controls should be assumed. Usually, harmonic controls (sinusoids with different frequencies) are selected. After that, the controls are substituted into the Lie series resulting from the gCBHD formula. Finally, having known required coefficients of the series, a set of algebraic equations on parameters of harmonics is obtained. Solving the set analytically (in simple cases) or numerically (in more demanding situations), controls that generate required terms are obtained.

#### 4. Quality measures to evaluate trajectories

In order to examine the influence of a representation on a trajectory leading to the goal configuration of the system (1), some evaluation tools are to be introduced. As the simplest trajectory joining the initial point with the target one is the straight line, it is natural to define measures relating an examined trajectory with this reference trajectory. It should be noted that for nonholonomic systems a straight line trajectory is hardly ever admissible, i.e., realizable with admissible controls.

Before defining some quality measures for trajectories, a preliminary step is performed. First, the shift of coordinates is performed  $\hat{\mathbf{q}} = \mathbf{q} - \mathbf{q}_0$  to convert the initial point of the planning  $\mathbf{q}_0$  into the origin of a new coordinate frame. Thus, the new target is  $\hat{\mathbf{q}}_f = \mathbf{q}_f - \mathbf{q}_0$  and points of trajectory are transformed appropriately. Then, canonical versors of the new coordinate frame  $\mathbf{e}_k^n$ ,  $k = 1, \dots, n$ , are rotated in such a manner that the new versor  $\hat{\mathbf{e}}_1^n$  is co-linear with  $\hat{\mathbf{q}}_f$  and equal to  $(1, 0, \dots, 0)$ , while the remaining versors  $\hat{\mathbf{e}}_k^n$ ,  $k = 2, \dots, n$  are selected according to the modified Gramm–Schmidt method (Golub and Loan, 1996). Initialized with the set of versors  $\{\hat{\mathbf{q}}_f / \|\hat{\mathbf{q}}_f\|, \mathbf{e}_2^n, \dots, \mathbf{e}_n^n\}$ , the Gramm–Schmidt algorithm produces an orthonormal basis  $\hat{\mathbf{e}}_k^n$ ,  $k = 1, \dots, n$ . In this basis the examined trajectory is expressed as  $\hat{\mathbf{q}}(t)$ ,  $t \in [0, T]$ , and its target point is  $\tilde{\mathbf{q}}_f = (\|\hat{\mathbf{q}}_f\|, 0, \dots, 0)$ .

In the new coordinate frame, the first coordinate naturally parametrizes a straight line motion toward the goal. When the coordinate transformations are performed in an obstacle cluttered environment, also obstacles are transformed appropriately to preserve their relationship with a trajectory. In a multidimensional case, having a small redundancy in selecting  $\hat{\mathbf{e}}_2^n$ , it may be desirable to point the versor toward the nearest obstacle from the line joining  $\mathbf{0}$  with  $\tilde{\mathbf{q}}_f$ .

To get an insight into the shape of the trajectory  $\hat{\mathbf{q}}(\cdot)$  in the new coordinate frame, two types of measures will be defined. The positional function measures the Euclidean distance of  $\hat{\mathbf{q}}(t)$  from the straight line,

$$\text{dist}(t) = \text{dist}(\hat{\mathbf{q}}(t), \text{line}(\mathbf{0}, \hat{\mathbf{q}}_f)) = \sqrt{\sum_{i=2}^n \hat{q}_i^2(t)}. \quad (13)$$

A fixed value of the measure (13) has a lot of points placed



on the  $(n - 1)$ -dimensional sphere. Therefore, an angular measure is to be defined to locate the point  $\hat{q}(t)$  more precisely. Because in the new coordinate frame versors  $\hat{e}_i^n$  are canonical, angles between  $(n - 1)$  versors and the vector  $\hat{q}(t)$  (the first angle is not considered as it does not carry any significant angular information),

$$\text{angle}_i(t) = \left\langle \frac{\hat{q}(t)}{\|\hat{q}(t)\|}, \mathbf{e}_{i+1}^n \right\rangle, \quad i = 1, \dots, n - 1. \quad (14)$$

Taking a reduced vector  $\tilde{q}_c$  as  $q_c$  with its first coordinate left, one can also use the generalized spherical coordinates  $(\theta_1, \dots, \theta_{n-1})$  as an angular measure. The spherical coordinates are related to the coordinates  $\tilde{q}_c = (q_{c2}, \dots, q_{cn})^T$  as follows:

$$\begin{aligned} q_{c2} &= \|\tilde{q}_c\| \cos \theta_1, \\ q_{ci} &= \|\tilde{q}_c\| \cos \theta_i \prod_{j=1}^{i-1} \sin \theta_j, \quad i = 3, \dots, n - 1, \\ q_{cn} &= \|\tilde{q}_c\| \prod_{j=1}^{n-1} \sin \theta_j. \end{aligned} \quad (15)$$

From Eqn. (15), the spherical coordinates can be retrieved according to the formula

$$\begin{aligned} \theta_{n-1} &= \text{atan2}(q_{cn}, q_{c(n-1)}), \\ \theta_i &= \text{atan2}(q_{c(i+1)}, q_{ci} \cos \theta_{i+1}), \quad i = n - 2, \dots, 1, \end{aligned} \quad (16)$$

where a standard numeric function  $\text{atan2}$  was used to avoid division by ‘zero’ while getting consecutive angles  $\theta_i$ . In the three dimensional space case  $n = 3$ , there is only one angle to determine, and the angular measures (14) and (16) are equivalent. The presented measures (13) and (14) evaluate the local aspect of a trajectory at a given time stamp. Based on the measures, a global behavior of a trajectory can be highlight. For the positional measure, two functions are defined. The first one,

$$\text{dist}_\Sigma = \int_0^T \text{dist}(t) dt, \quad (17)$$

averages the trajectory, while the second,

$$\text{dist}_{\max} = \max_{t \in [0, T]} \text{dist}(t), \quad (18)$$

checks its extremal amplitude over the time horizon.

## 5. Simulation results

In the first experiment, the LS algorithm was applied to compute trajectories of the Brockett integrator described by the equations

$$\dot{q} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \mathbf{X}(q)u_1 + \mathbf{Y}(q)u_2 = \begin{bmatrix} 1 \\ 0 \\ -q_2 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 1 \\ q_1 \end{bmatrix} u_2. \quad (19)$$

Table 3. P.H. representations of motion (for  $m = 2$  and order of nilpotency 2) and the types of their CFS equations.

No.	P.H. representation, $\mathcal{S}(t)$	type
1	$e^{h_1 X + h_2 Y + h_3 [X, Y]}$	#1
2	$e^{h_1 X + h_2 Y} e^{h_3 [X, Y]}$	#1
3	$e^{h_3 [X, Y]} e^{h_1 X + h_2 Y}$	#1
4	$e^{h_1 X + h_3 [X, Y]} e^{h_2 Y}$	#2
5	$e^{h_2 Y} e^{h_1 X + h_3 [X, Y]}$	#3
6	$e^{h_2 Y + h_3 [X, Y]} e^{h_1 X}$	#3
7	$e^{h_1 X} e^{h_2 Y + h_3 [X, Y]}$	#2
8	$e^{h_1 X} e^{h_2 Y} e^{h_3 [X, Y]}$	#2
9	$e^{h_1 X} e^{h_3 [X, Y]} e^{h_2 Y}$	#2
10	$e^{h_2 Y} e^{h_1 X} e^{h_3 [X, Y]}$	#3
11	$e^{h_2 Y} e^{h_3 [X, Y]} e^{h_1 X}$	#3
12	$e^{h_3 [X, Y]} e^{h_1 X} e^{h_2 Y}$	#2
13	$e^{h_3 [X, Y]} e^{h_2 Y} e^{h_1 X}$	#3

The motion planning task was to move the state from the initial configuration  $q_0 = (0, 0, 0)^T$  to the goal one  $q_f = (1, 2, 3.2)^T$  in the time horizon  $T = 1$ . The Brockett integrator is an example of a controllable nilpotent system with the order of nilpotency equal to 2, thus  $r = 3$ . According to Table 2, 13 motion representations, gathered in Table 3, can be defined for the system (19). For each representation, the corresponding Chen–Fliess–Sussmann equation described by (11) was calculated. The equations can be divided into three groups (enumerated as in Table 3),

$$\begin{aligned} &\text{type \#1} \\ &\begin{cases} \dot{h}_1 = v_1, \\ \dot{h}_2 = v_2, \\ \dot{h}_3 = \frac{1}{2}(h_1 v_2 - h_2 v_1) + v_3, \end{cases} \end{aligned}$$

$$\begin{aligned} &\text{type \#2} && \text{type \#3} \\ &\begin{cases} \dot{h}_1 = v_1, \\ \dot{h}_2 = v_2, \\ \dot{h}_3 = -h_2 v_1 + v_3, \end{cases} && \begin{cases} \dot{h}_1 = v_1, \\ \dot{h}_2 = v_2, \\ \dot{h}_3 = h_1 v_2 + v_3, \end{cases} \end{aligned}$$

all initialized with  $h(0) = \mathbf{0}$ . In the last step of the LS algorithm, for a given representation (with its parameters  $h_i(T)$ ,  $i = 1, \dots, r$ , already determined) the gCBHD formula (Duleba and Khefifi, 2006) was applied to generate controls. In each of  $\bar{r}$  time intervals of the length  $T' = T/\bar{r}$ , the controls were taken in the form  $u_1(s) = p_1 + p_2 \sin(2\pi s/T')$ ,  $u_2(s) = p_3 + p_4 \cos(2\pi s/T')$ , where  $s \in [0, T']$  and  $\mathbf{p} = (p_1, \dots, p_4)^T$  were parameters to be found. The parameters were computed using a version of the Newton algorithm to solve the inverse task  $\mathbf{f}(\mathbf{p}) = \mathbf{h}(T)$  with  $\mathbf{h}(T)$  known and the function  $\mathbf{f}(\mathbf{p})$  determined by the gCBHD formula for given controls.

In Fig. 2, graphs of the distance measure (13) were

presented for different representations (no angle characteristics were drawn as being less informative than positional ones). In each figure, a family of characteristics was shown with the goal point given as  $\mathbf{q}_f(\xi) = \mathbf{q}_0 + \xi(\mathbf{q}_f - \mathbf{q}_0)$  with  $\xi \in (0, 1]$  varied. In Table 4, values of the global positional measures (17) and (18) were collected, while in Fig. 1, control signals were depicted for selected representations. It is clearly visible, cf. Fig. 2, that the selection of a motion representation plays a key role in shaping a trajectory directed to the goal point. For the selected set of boundary points, the best representation (#1) gives the maximal distance from the straight line  $\mathbf{q}_0\mathbf{q}_f$  about 15 times smaller than the worst representation (#10).

Frequently, for a single representation there are many optima in function  $\text{dist}(t)$  and the maximal values are attained for various  $t$  (so at different locations along the line joining  $\mathbf{q}_0$  with  $\mathbf{q}_f(\xi)$ ). From Fig. 2 and Table 4 one can conclude that trajectories (for any representation) have smaller and smaller amplitudes when the goal point is closer and closer to the initial one. This fact is crucial in using the LS algorithm in an obstacle cluttered environment. This means that if only any trajectory (not necessarily realizable with admissible controls) obtained with a holonomic method (potential field method, Voronoi diagram or any other) is open (i.e., its closest neighborhood is composed of obstacle-free points only), the trajectory can be approximated (realized) with the nonholonomic LS method and this trajectory will also be obstacle free.

The controls shown in Fig. 1 are composed of one, two and three equi-length time intervals (their number equals  $\bar{r}$ ), and their amplitudes differ significantly. It appears that representations with a smaller number of segments are more energy efficient than those with more segments. One can also conclude that the first, canonical representation seems to be best within the whole family of representations. Moreover, representations found in the robotic literature (forward and backward) are likely among the worst ones as they are composed of the maximal number of control segments.

Results of the sample task provided visual information on differences among the representations. However, the task cannot be considered an attributive one. Therefore, a set of tasks were run to gain statistically relevant data. The initial state for all tasks was set to  $(0, 0, 0)^T$  and a time horizon  $T = 1$  was fixed. In spherical coordinates  $(R, \alpha, \beta)^T$ , goal points were located on a sphere with the radius  $R = 1$  and  $\alpha \in \{i \cdot 10^\circ\}$ ,  $i = 0, \dots, 35$ ,  $\beta \in \{j \cdot 10^\circ\}$ ,  $j = -9, \dots, 9$ . Two factors were evaluated: an energy expenditure on controls  $\text{energy}(\mathbf{u}(\cdot)) = \int_{t=0}^T (u_1^2 + u_2^2) dt$  and the maximal distance  $d_{\max}$  given by Eqn. (18).

For all 13 representations, a mean value (mean =  $\sum_{i=1}^s v_i/s$ ) and an unbiased standard deviation (std dev =  $\sqrt{\sum_{i=1}^s (v_i - \text{mean})^2/(s - 1)}$ ) of an

evaluated factor with its data gathered from the tasks  $\mathbf{v} = (v_1, \dots, v_s)^T$  are presented in Table 5. It appears that observations made for the single task are also valid for a set of tasks. Moreover, relatively large values of the standard deviation of the energy show that there are energy cheap and expensive directions of motion.

The last example will cover a four dimensional controllable, nilpotent chain system with two inputs and order of nilpotency equal to 3. The system is described by the equations

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \end{bmatrix} = \mathbf{X}(\mathbf{q})u_1 + \mathbf{Y}(\mathbf{q})u_2 = \begin{bmatrix} 1 \\ 0 \\ q_2 \\ q_3 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u_2. \tag{20}$$

Besides  $\mathbf{X}, \mathbf{Y}$ , only vector fields  $[\mathbf{X}, \mathbf{Y}] = (0, 0, -1, 0)^T$ ,  $[\mathbf{X}, [\mathbf{X}, \mathbf{Y}]] = (0, 0, 0, 1)^T$  do not vanish. In particular,  $[\mathbf{Y}, [\mathbf{X}, \mathbf{Y}]] = (0, 0, 0, 0)^T$ . As too many representations

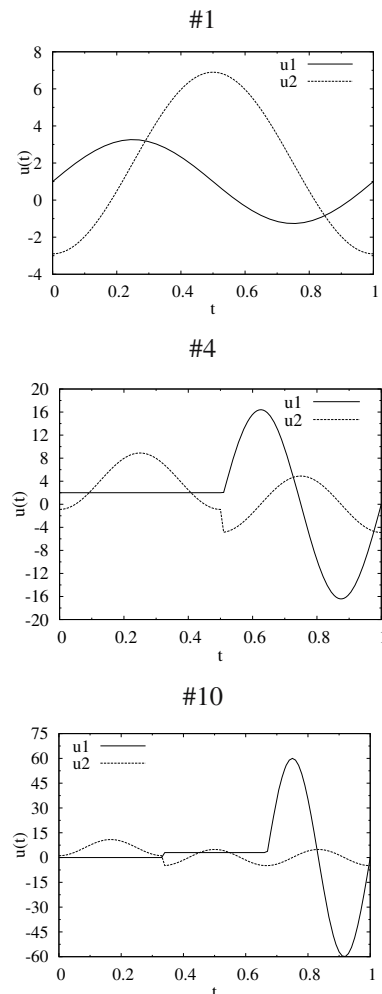


Fig. 1. Controls to reach the goal  $\mathbf{q}_f$  for selected representations.

Table 4. Values of positional measures for the goal point (controlled by the coefficient  $\xi$ ) and a motion representation varied.

rep no.	Positional measure (17)					Positional measure (18)				
	$\xi$					$\xi$				
	0.2	0.4	0.6	0.8	1.0	0.2	0.4	0.6	0.8	1.0
1	0.40	0.40	0.41	0.43	0.45	0.65	0.65	0.68	0.73	0.82
2	0.35	0.68	1.14	1.75	2.52	0.64	1.65	3.12	5.14	7.80
3	0.30	0.45	0.56	0.64	0.69	0.56	0.90	1.15	1.30	1.33
4	0.28	0.38	0.47	0.54	0.61	0.50	0.75	0.92	1.03	1.12
5	0.26	0.37	0.51	0.69	0.91	0.40	0.63	0.87	1.25	1.76
6	0.37	0.68	1.05	1.49	1.99	0.68	1.56	2.64	3.94	5.46
7	0.31	0.53	0.76	1.01	1.26	0.59	0.97	1.42	1.88	2.35
8	0.32	0.55	0.74	0.88	0.95	0.84	1.64	2.31	2.77	2.90
9	0.30	0.48	0.64	0.76	0.86	0.82	1.45	1.91	2.19	2.90
10	0.41	0.90	1.53	2.34	3.34	1.08	2.73	5.09	8.32	12.53
11	0.31	0.63	1.08	1.68	2.45	0.86	2.24	4.26	7.06	10.78
12	0.30	0.48	0.60	0.66	0.67	0.64	1.07	1.34	1.43	1.34
13	0.33	0.67	1.10	1.61	2.20	0.81	1.78	2.93	4.26	5.76

Table 5. Statistical evaluation of 13 representations with respect to the energy expenditure and the maximal distance factors for the Brockett integrator.

rep. no.	total energy		maximal dist.	
	mean	std. dev.	mean	std. dev.
1	4.96	1.71	0.50	0.12
2	11.46	2.20	0.67	0.13
3	11.44	2.19	0.62	0.15
4	13.04	2.66	0.71	0.16
5	13.05	2.66	0.65	0.12
6	10.25	3.29	0.61	0.10
7	10.25	3.29	0.60	0.11
8	19.85	4.29	0.73	0.14
9	19.85	4.29	0.76	0.14
10	19.85	4.29	0.72	0.14
11	19.85	4.29	0.74	0.11
12	19.85	4.29	0.69	0.13
13	19.85	4.29	0.68	0.12

are available for this case, cf. Table 2, only three representations, namely, canonical, forward and backward ones, were evaluated. CFS equations for the representations are the following:

- the canonical representation:

$$\begin{aligned}
 & (\dot{h}_1, \dot{h}_2, \dot{h}_3, \dot{h}_4, \dot{h}_5)^T \\
 &= (v_1, v_2, \frac{1}{2}(h_1 v_2 - h_2 v_1) + v_3, \\
 & \quad v_4 + \frac{1}{12}h_1(h_1 v_2 - h_2 v_1) + \frac{1}{2}(h_1 v_3 - h_3 v_1), \\
 & \quad v_5 + \frac{1}{12}h_2(h_1 v_2 - h_2 v_1) + \frac{1}{2}(h_2 v_3 - h_3 v_2))^T,
 \end{aligned}$$

- the forward representation:

$$\begin{aligned}
 & (\dot{h}_1, \dot{h}_2, \dot{h}_3, \dot{h}_4, \dot{h}_5)^T \\
 &= (v_1, v_2, v_3 - h_2 v_1, v_4 - h_3 v_1, \\
 & \quad v_5 - h_3 v_2 + \frac{1}{2}h_2^2 v_1)^T,
 \end{aligned}$$

- the backward representation:

$$\begin{aligned}
 & (\dot{h}_1, \dot{h}_2, \dot{h}_3, \dot{h}_4, \dot{h}_5)^T \\
 &= (v_1, v_2, v_3 + h_1 v_2, \\
 & \quad v_4 + h_1 v_3 + \frac{h_1^2}{2}v_2, v_5 + h_2 v_3 + h_1 h_2 v_2)^T.
 \end{aligned}$$

It is worth noticing that, despite the fact that the chain system (20) is four dimensional,  $n = 4$ , the corresponding CFS equations are five dimensional. It is quite clear, cf. Eqn. (12), that CFS equations have to cover whole layers of vector fields as, in a general case, any vector field appearing in a given layer has a comparable impact on the resulting motion to any other vector field from the same layer. For the system (20), the  $h_5$  term does not impact the motion as it multiplies the vanishing vector field  $[Y, [X, Y]]$ .

To generate  $\mathbf{h}(T)$ , in each interval  $i = 1, \dots, \bar{\tau}$  controls were selected from the family  $u_1(s) = p_1 + p_2 \sin(\omega s) + p_3 \sin(2\omega s)$ ,  $u_2(s) = p_4 + p_5 \cos(\omega s) + p_6 \cos(2\omega s)$ , where  $s \in [0, T/\bar{\tau} = T']$ ,  $\omega = 2\pi/T'$ . As in the previous example, the values of the parameters  $\mathbf{p} = (p_1, \dots, p_6)^T$  were determined with the use of the Newton algorithm, although the control-dependent function  $\mathbf{f}(\mathbf{p})$  is quite different in this case. Simulations were carried out on a set of tasks with uniformly distributed goal points placed on a surface of the four dimensional sphere with the radius  $R = 1$  (more than 300 tasks were

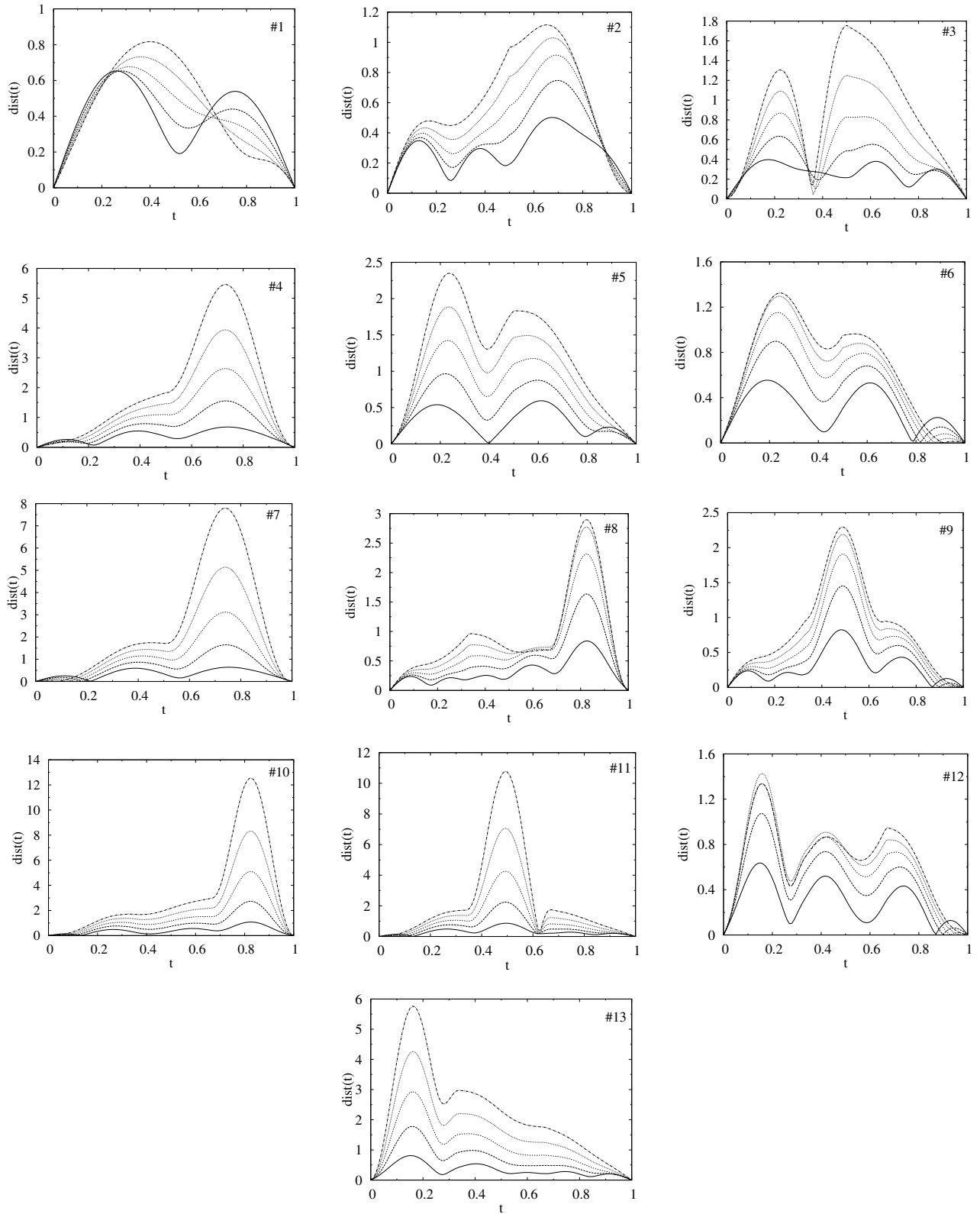


Fig. 2. Distance of  $q(t)$  from the straight line  $q_0q_f$  as a function of  $t$  for a family of goals controlled by the coefficient  $\xi = 0.2 \cdot i$ ,  $i = 1, \dots, 5$  and various representations. In almost any case, curves corresponding to a smaller value of  $i$  are placed below those with higher  $i$ .



generated). The initial configuration for all the tasks was  $(0, 0, 0, 0)^T$  and the time horizon  $T = 1$ . Also, for the four dimensional case, the observations made for the three dimensional Brockett integrator remain valid.

While presenting and comparing representations, their computational complexity should be addressed. In many cases, it is not a crucial problem in motion planning because the planning can be performed in the off-line mode, and a resulting trajectory or path supplied to a control module (which works in the on-line mode) is approximated as accurately as possible. To compare the computational complexity of representations, let us note that for a given representation its CFS differential equation can be computed in the off-line mode as it does not depend on a particular configuration of a system.

Computing  $\mathbf{h}(T)$  needs approximately the same amount of time for any representation as it requires integration of the CFS equation corresponding to the representation for given  $\mathbf{v}(\cdot)$ . The only difference in complexity between representations is in determining controls to generate  $\mathbf{h}(T)$ . Clearly, the complexity is more or less proportional to the number of segments  $r$ . In this aspect the canonical representation is best ( $\bar{r} = 1$ ) while forward and backward representations are worst ( $\bar{r} = r \geq n$ ). However, for forward and backward representations, motions along segments  $e^{h_i X}$ ,  $e^{h_j Y}$  always appear, and they can be computed without using the Newton algorithm, just by switching one control and switching off the other. For the canonical representation the easy-to-generate segments never appear.

Another interesting question to be answered concerns the best representation among those possible to be generated. According to the presented simulation results, it seems that the canonical representation is the best one from among all other representations, at least for obstacle free environments. However, from the perspective of avoiding obstacles, representations are much harder to compare. Trajectories joining two fixed boundary points and corresponding to various motion representations likely pass through different areas of a state space, cf. Fig. 2. In this sense any representation can be made the best one from among all considered by a simple construction.

Let a trajectory be generated for a given representation and fixed boundary points in an obstacle free environ-

ment. Now, make the trajectory surrounded with a small radius tube. Only the points inside the tube are obstacle free for a next task to be performed. It is easy to observe that any other representation will not be as effective in planning the desired motion as that one just considered because, likely, the other representations will generate trajectories colliding with obstacles. In practice there is no other way to evaluate representations with respect to collision avoidance than to generate their corresponding trajectories and check their distances from the nearest obstacle. Fortunately, it is a simple numerical integration procedure which can be immediately abandoned when a collision is detected.

## 6. Conclusions

In this paper the LS algorithm for motion planning of driftless nilpotent systems was extended by using a variety of motion representations. In a systematic way the representations were introduced and exemplified on controlling the Brockett integrator and the four dimensional chain system. Some positional and angular measures were proposed to relate a given trajectory with a straight-line trajectory joining boundary points of the planning. It was shown that motion representations have great impact on evolution of trajectories joining fixed boundary points. Consequently, by changing the representation it is possible either to optimize the shape of a trajectory towards the goal point or to avoid obstacles on its way. Some remarks on the computational complexity of processing the representations were provided. In the future, the developments of this paper are to be applied to plan a motion of real robots (via local transformations their models can be approximated with nilpotent systems).

## Acknowledgment

The work of the second author is supported through funding for Polish science for the years 2010–2011 within a research project.

## References

- Bellaïche, A., Laumond, J.-P. and Chyba, M. (1993). Canonical nilpotent approximation of control systems: Application to nonholonomic motion planning, *IEEE Conference on Decision and Control, San Antonio, TX, USA*, pp. 2694–2699.
- Chow, W.L. (1939). On system of linear partial differential equations of the first order, *Mathematische Annalen* **117**(1): 98–105, (in German).
- Dulęba, I. (1998). *Algorithms of Motion Planning for Nonholonomic Robots*, Wrocław University of Technology Publishing House, Wrocław.
- Dulęba, I. (2009). How many P. Hall representations there are for motion planning of nilpotent nonholonomic systems?.

Table 6. Statistical evaluation of three representations with respect to energy expenditure and the maximal distance factors for the four dimensional chain system.

representation	total energy		maximal dist.	
	mean	std. dev.	mean	std. dev.
backward	924.54	359.57	2.99	0.35
forward	965.49	364.77	3.26	0.42
canonical	68.49	63.29	2.17	0.96

- IFAC International Conference on Methods and Models in Automation and Robotics, Międzyzdroje, Poland.*
- Dułęba, I. and Jagodziński, J. (2008a). On impact of reference trajectory on Lafferriere–Sussmann algorithm applied to the Brockett integrator system, in K. Tchoń and C. Zielinski (Eds.), *Problems of Robotics*, Science Works: Electronics, Vol. 166, Warsaw University of Technology Publishing House, Warsaw, pp. 515–524, (in Polish).
- Dułęba, I. and Jagodziński, J. (2008b). On the structure of Chen–Fliess–Sussmann equation for Ph. Hall motion representation, in K. Tchoń (Ed.), *Progress in Robotics*, Wydawnictwa Komunikacji i Łączności, Warsaw, pp. 9–20.
- Dułęba, I. and Jagodziński, J. (2009). Computational algebra support for the Chen–Fliess–Sussmann differential equation, in K. Kozłowski (Ed.), *Robot Motion and Control*, Lecture Notes in Control and Information Sciences, Vol. 396, Springer, Berlin/Heidelberg, pp. 133–142.
- Dułęba, I. and Khefifi, W. (2006). Pre-control form of the generalized Campbell–Baker–Hausdorff–Dynkin formula for affine nonholonomic systems, *Systems and Control Letters* **55**(2): 146–157.
- Golub, G.H. and Loan, C.V. (1996). *Matrix Computations*, 3rd Edn., The Johns Hopkins University Press, London.
- Koussoulas, N. and Skiadas, P. (2001). Motion planning for drift-free nonholonomic system under a discrete levels control constraint, *Journal of Intelligent and Robotic Systems* **32**(1): 55–74.
- Koussoulas, N. and Skiadas, P. (2004). Symbolic computation for mobile robot path planning, *Journal of Symbolic Computation* **37**(6): 761–775.
- Lafferriere, G. (1991). A general strategy for computing steering controls of systems without drift, *IEEE Conference on Decision and Control*, Brighton, UK, pp. 1115–1120.
- Lafferriere, G. and Sussmann, H. (1990). Motion planning for controllable systems without drift: A preliminary report, *Technical report*, Rutgers Center for System and Control, Piscataway, NJ.
- Lafferriere, G. and Sussmann, H. (1991). Motion planning for controllable systems without drift, *IEEE Conference on Robotics and Automation*, Brighton, UK, pp. 1148–1153.
- LaValle, S. (2006). *Planning Algorithms*, Cambridge University Press, Cambridge, MA.
- Reutenauer, C. (1993). *Free Lie Algebras*, Clarendon Press, Oxford.
- Rouchon, P. (2001). Motion planning, equivalence, infinite dimensional systems, *International Journal of Applied Mathematics and Computer Science* **11**(1): 165–188.
- Strichartz, R.S. (1987). The Campbell–Baker–Hausdorff–Dynkin formula and solutions of differential equations, *Journal of Functional Analysis* **72**(2): 320–345.
- Struemper, H. (1998). Nilpotent approximation and nilpotentization for under-actuated systems on matrix Lie groups, *IEEE Conference on Decision and Control*, Tampa, FL, USA, pp. 4188–4193.
- Sussmann, H. (1991). Two new methods for motion planning for controllable systems without drift, *European Control Conference*, Grenoble, France, pp. 1501–1506.
- Vendittelli, M., Oriolo, G., Jean, F. and Laumond, J.-P. (2004). Nonhomogeneous nilpotent approximations for nonholonomic system with singularities, *IEEE Transactions on Automatic Control* **49**(2): 261–266.



**Ignacy Dułęba** was born in 1961. He received M.Sc. and Ph.D. degrees in robotics from the Wrocław University of Technology in 1986 and 1991, respectively. In 1999 he received his D.Sc. degree from the Warsaw University of Technology. Currently, he is a professor at the Wrocław University of Technology. His scientific activities have been concerned with the modelling of robots, motion planning of nonholonomic systems, and control.



**Jacek Jagodziński** was born in 1983. He obtained his M.Sc. degree in control engineering and robotics from the Wrocław University of Technology, Poland, in 2007. Currently, he is working towards his Ph.D. thesis devoted to analytical methods of motion planning. His research interests cover the area of control theory applied to nonholonomic robotic systems.

Received: 31 May 2010

Revised: 28 September 2010