BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 71(3), 2023, Article number: e145684
DOI: 10.24425/bpasts.2023.145684

Sub-optimal motion planning of one-chained, two-input nonholonomic systems

Ignacy DULEBA[1]* **and Iwona KARCZ-DULEBA**[2]

[1] Department of Cybernetics and Robotics
[2] Department of Control Systems and Mechatronics
Wroclaw University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

Abstract. In this paper three algorithms of motion planning for two-input, one-chained nonholonomic systems are presented. The classical Murray-Sastry algorithm is compared with two original algorithms aimed at optimizing energy of controls. Based on the generalized Campbell-Baker-Hausdorff-Dynkin formula applied to the systems, some observations are made concerning the optimal relationship between amplitudes and phases of harmonic controls. The observations help to optimize a selection of controls and to design new algorithms for planning a sub-optimal trajectory between given boundary configurations. It was also shown that for those particular systems the generalized C-B-H-D formula is valid not only locally (as in a typical case) but also globally. Simulations performed on the five-dimensional chain system facilitate distinguishing the proposed algorithms from the Murray-Sastry algorithm and to illustrate their features. Systems in a chained form are important from a practical point of view as they are canonical for a class of systems transformable into this form. The most prominent among them are mobile robots with or without trailers.

Key words: nonholonomic systems; chained form; optimization; algorithm; Lie algebraic method.

1. INTRODUCTION

Off-line motion planning of robotic systems is aimed at preparing a desired trajectory (path) to be executed in on-line mode at a control stage. In the robotic literature, there are many methods and techniques developed to solve the task [1] that depends on a model of the robot and its environment. Local methods achieve the goal of planning via a series of sub-plannings where an end-state of the current planning becomes initial one for the next planning. Local methods are computationally not so involved as global ones that plan a whole motion at once (but frequently in an iterative manner similarly to local methods). Moreover, local methods are not so sensitive to changes in robot environment. Methods can be divided with respect to a quality of obtained solutions (not optimized, sub- or optimal ones) and how they incorporate extra constraints (on controls, state, obstacle avoidance) – explicit constraints or introduced via penalty into a cost function [2]. A model of a robot plays a crucial role in planning algorithms [3, 4]. Discrete models tend to consider a robot as an automaton that changes its state at some asynchronous time stamps while continuous models describe a robot with differential equations. Continuous models frequently result from constraints imposed on a robot (for example in the Pfaff form like for simple mobile robots: the unicycle or the kinematic car [1] or tractor pulling some trailers, or free-floating robots [3]). When the constraints are holonomic one, a dimension of a configuration space can be reduced by the number of constraints while for nonholonomic systems the constraints are not integrable and resulting systems (despite being controllable) are difficult to control because a number of controls is smaller than a configuration space dimension. Continuous models can be considered at the kinematic level and described by first-order (nonlinear) differential equations or at the dynamic level when also torques and accelerations are coupled in the second-order equations.

The next dichotomy results from the number of systems to which a planning method can be applied. In this scope we distinguish general purpose methods [2, 5, 6] and dedicated ones that work only for a specific model [7, 8] or a small number of models [9]. General-purpose methods are computationally involved, solved or supported with an extensive usage of numerical methods and some numerical problems in high dimensional search spaces are to be expected. Difficulty grows rapidly as optimization of a cost function is to be performed. Although the optimal control theory based on the Pontriagin Maximum Principle or Bellman equations is well established [10] still very difficult tasks are generated (the two-point boundary value problem for continuous tasks or solved with the dynamic programming technique in a discrete setting). A very popular technique in solving the optimal control tasks for general systems is rooted in a continuation method [11]. It relies on a modification of the current best trajectory (solution) based on a linear approximation of the system along this trajectory. Unfortunately, its efficiency strongly depends on an initial trajectory. When implemented as a local steepest-descent method the current trajectory can easily be trapped at a local optimum. Partially this disadvantage can be weakened by using a multi-start technique but computational effort is increased. Dedicated methods can be

*e-mail: ignacy.duleba@pwr.edu.pl

Manuscript submitted 2022-12-28, revised 2023-03-08, initially accepted for publication 2023-03-27, published in June 2023.

suited into a particular system structure and use it to simplify a task to be solved. Sometimes even analytical solutions [7] can be obtained with the methods.

Motion planning tasks can be motivated by practical necessity to control real objects or just a mathematical curiosity [7]. Recently numerous motion planning methods are based on graph-searching techniques able to find desired trajectories in obstacle-clutter environments. The most popular among them are the $A^\star$ algorithm, rapidly exploring random trees, [12], probabilistic roadmaps [13], or their numerous versions. Frequently the techniques assume planning for a holonomic (omnidirectional) robot although some can also use continuous models to find local paths between neighbor nodes while creating a graph. Graph-based techniques dominate in motion planning for multi-robot systems.

A very general strategy to deal with a continuous planning task is to transform uniquely a given system into a simple (canonical) form, solve the task and move back results into the original space. In this case solving one problem given in a canonical form, solves automatically quite a lot of transformable problems. Systems in a canonical form for a sub-class of nonholonomic ones have been proposed [14] and algorithms to transform them into this form developed [15]. Contact structures, one- or two-chained systems are just examples of systems in a canonical form. In robotics the chained systems frequently appear when mobile robots with or without trailers are analyzed [16].

When a new class of systems is defined, a few tendencies are usually observed. The class is being extended to cover more systems [17] or original systems are varied (to cover inaccuracies in modeling, to optimize their behavior, to consider extra constraints [18]) or the systems form a base for defining related tasks like control (tracking) problems when a desired trajectory is given [19]. In this paper energy optimization for one-chained systems is of the main concern. Since the seminal Brockett work [14], it is known that optimal controls of chained systems are to be searched within a mixture of harmonic functions. It is a hard computational problem to find the optimal solution within this class of controls. Murray and Sastry [9, 20] proposed a much simpler approach where each control is just described by a single sinusoidal function. Based on the special form of canonical systems, clearly visible from a Lie algebraic perspective, effective steering methods based on sinusoidal controls were proposed [21]. With appropriately selected parameters of controls, consecutive coordinates of a configuration vector are brought to their desired values, while circling loops in coordinates already steered. Unfortunately, the Murray-Sastry algorithm does not admit energy optimization.

In this paper energy sub-optimal controls are searched for one-chained systems with two controls and presented within the framework of Lie algebraic methods which proved to be appropriate to plan a motion of general driftless systems [22] or tracing their desired trajectories [23]. In Section 2 the generalized Campbell-Hausdorff-Dynkin (gCBHD) formula is presented in the context of one-chained systems. Although the formula is used mainly in a local nonholonomic motion planning [24] but it is quite general and solutions of global and local methods should be consistent when boundary configurations are close to each other. Using the gCBHD formula, it is shown how to generate control-dependent coefficients of vector fields required to satisfy a controllability condition and to express them in a close form as a function of parameters of controls. Afterwards, applying the Lagrange multiplier technique, the parameters are selected optimally. In Section 3 a basic algorithm of motion planning with sinusoidal controls is recalled, then a few modifications – algorithms are constructed according to rules derived from the gCBHD formula. In Section 4 simulation results are provided for a two-input one-chained system with a five-dimensional configuration space. Section 5 concludes the paper.

According to the classifications mentioned in this section the method is designed for driftless, also high-dimensional, nonholonomic systems expressed in a canonical form. A motion planning task is solved with a dedicated, analytical algorithm able to minimize energy of controls. In the presented version collision-free environment is assumed. However, this condition can be released when appropriate sequence of sub-goals is defined to avoid possible collisions with obstacles. Some hints how to plan a motion of chained systems among obstacles are presented in [25].

2. A LIE ALGEBRAIC CHARACTERIZATION OF ONE-CHAINED SYSTEMS WITH THE GCBHD FORMULA

The gCBHD formula describes (locally) a trajectory of a non-autonomous system of differential equations initialized at a given configuration

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{A}(t)(\boldsymbol{q}(t)) \quad \{= \boldsymbol{X}(\boldsymbol{q})u_1 + \boldsymbol{Y}(\boldsymbol{q})u_2\}, \quad \boldsymbol{q}(0) = \boldsymbol{q}_0, \quad (1)$$

where $\boldsymbol{A}(t)(\cdot)$ is a family of analytic vector fields parameterized continuously by t. In equation (1) two-input driftless systems, considered in this paper, are distinguished by curly parentheses. The systems are defined by vector fields $\boldsymbol{X}(\boldsymbol{q}), \boldsymbol{Y}(\boldsymbol{q})$ called generators. A solution to equation (1) has the form

$$\boldsymbol{q}(t) = \exp \boldsymbol{z}(t)(\boldsymbol{q}(0)) \simeq \boldsymbol{z}(t) + \boldsymbol{q}(0), \quad (2)$$

where $\exp \boldsymbol{z}(t)(\boldsymbol{q}_0)$ is a solution of the equation

$$d\boldsymbol{v}(s,t)/ds = z(t)\boldsymbol{v}(s)$$

with the boundary condition $\boldsymbol{v}(0,0) = \boldsymbol{q}_0$.

$$\boldsymbol{v}(s,t) = \exp(s\boldsymbol{z}(t))(\boldsymbol{q}_0) \;\Rightarrow\; \boldsymbol{q}(t) = \boldsymbol{v}(1,t) = \exp \boldsymbol{z}(t)(\boldsymbol{q}_0).$$

For $t \to 0$, $\boldsymbol{z}(t)$ takes a form of the series [26]:

$$\boldsymbol{z}(t) \sim \sum_{r=1}^{\infty} \sum_{\sigma \in P_r} \frac{(-1)^{e(\sigma)}}{r^2 \binom{r-1}{e(\sigma)}} \cdot \int_{T_r(t)} \left[\ldots [\boldsymbol{A}(s_{\sigma(1)}), \boldsymbol{A}(s_{\sigma(2)})] \ldots \boldsymbol{A}(s_{\sigma(r)}) \right] \mathrm{d}s^r, \quad (3)$$

where abbreviated notations are used

$$\int\limits_{T_r(t)} = \int\limits_{s_r=0}^{t} \int\limits_{s_{r-1}=0}^{s_r} \dots \int\limits_{s_1=0}^{s_2}, \quad \mathrm{d}s^r = \mathrm{d}s_1\,\mathrm{d}s_2 \dots \mathrm{d}s_r. \tag{4}$$

P_r denotes all permutations of the set: $\{1,\dots,r\}$; $e(\sigma\in P_r)$ is the number of errors in the permutation: $\sigma = \{\sigma(1),\sigma(2),\dots,\sigma(r)\}$, with the error increased by one each time when the next element in the permutation is smaller than the current one. In coordinates, the Lie bracket $[\cdot,\cdot]$, extensively used in (3), is defined as

$$[\boldsymbol{V},\boldsymbol{Z}] = \frac{\partial \boldsymbol{Z}}{\partial \boldsymbol{q}}\boldsymbol{V} - \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{q}}\boldsymbol{Z}. \tag{5}$$

Despite being much more complicated, the gCBHD formula plays the same role in differential equations as the Taylor series expansion for functions. It allows us to predict behavior of the system in a close vicinity of a given configuration based on characteristics of generator vector fields (and their descendants) at the configuration. The gCBHD formula can be interpreted as a shift operator which translates a current state $\boldsymbol{q}_0$ of system (1) to the position $\boldsymbol{q}_0 + \boldsymbol{z}(t)(\boldsymbol{q}_0)$. While processing formula (3), a lot of Lie brackets should be computed. In order to simplify the computations and to get a minimal number of terms, some properties of Lie brackets are used (bi-linearity and the Jacobi identity). Thus formula (3) is expressed as a series of independent elements of a basis (the Ph. Hall basis is frequently used) multiplied with control-dependent coefficients. Elements of the basis, initialized with $\boldsymbol{X}$, $\boldsymbol{Y}$ generators (1) in two-input case, can be ordered into layers. Each layer is composed of fixed degree elements and a degree denotes how many generators are used to describe a given element, for example $\deg(\boldsymbol{X}) = 1$, $\deg([\boldsymbol{Y},[\boldsymbol{X},\boldsymbol{Y}]) = 3$. The Ph. Hall basis up to the fourth layer spanned by generators $\boldsymbol{X},\boldsymbol{Y}$ is the following

$$\begin{aligned}&\boldsymbol{X},\boldsymbol{Y},[\boldsymbol{X},\boldsymbol{Y}],[\boldsymbol{X},[\boldsymbol{X},\boldsymbol{Y}]],[\boldsymbol{Y},[\boldsymbol{X},\boldsymbol{Y}]],\\ &[\boldsymbol{X},[\boldsymbol{X},[\boldsymbol{X},\boldsymbol{Y}]]],[\boldsymbol{Y},[\boldsymbol{X},[\boldsymbol{X},\boldsymbol{Y}]]],[\boldsymbol{Y},[\boldsymbol{Y},[\boldsymbol{X},\boldsymbol{Y}]]].\end{aligned}$$

Now, the gCBHD formula will be applied to two-input, one chained systems described in the n-dimensional configuration space by equations

$$\dot{\boldsymbol{q}}(t) = \begin{bmatrix}\dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\\ \dots\\ \dot{q}_n\end{bmatrix} = \begin{bmatrix}1\\ 0\\ q_2\\ \dots\\ q_{n-1}\end{bmatrix} u_1 + \begin{bmatrix}0\\ 1\\ 0\\ \dots\\ 0\end{bmatrix} u_2 = \boldsymbol{X}(\boldsymbol{q})u_1 + \boldsymbol{Y}(\boldsymbol{q})u_2. \tag{6}$$

Calculating Lie brackets (5) for generators $\boldsymbol{X},\boldsymbol{Y}$ and their descendants, it can be noticed that only non-vanishing brackets are of the form $ad^r_{\boldsymbol{X}}\boldsymbol{Y}$, $r = 1,\dots,n-2$, (where $ad^r_{\boldsymbol{X}}\boldsymbol{Y} = [\boldsymbol{X}, ad^{r-1}_{\boldsymbol{X}}\boldsymbol{Y}]$ and $ad^0_{\boldsymbol{X}}\boldsymbol{Y} = \boldsymbol{Y}$) and equal to

$$ad^r_{\boldsymbol{X}}\boldsymbol{Y} = (0,\dots,0,(-1)^r,0,\dots,0)^T, \tag{7}$$

with the value ± 1 placed at the $(r+2)$-nd coordinate. Thus in each layer of vector fields with degree at least two, there is only one nonzero vector field. Moreover, it is independent of all previous ones and according to Chow theorem [27] the system is nonholonomic with degree of nonholonomy equal to $(n-2)$. Taking advantage of enormous decrease in a number of items in the Ph. Hall basis in this case, a general formula for $\boldsymbol{z}(t)$ taken from [28] gets its simplified form

$$\begin{aligned}\boldsymbol{z}(t) \simeq{}& \tilde{u}_1\boldsymbol{X} + \tilde{u}_2\boldsymbol{Y} + \frac{1}{2}(\tilde{u}_{12} - \tilde{u}_{21})\,ad^1_{\boldsymbol{X}}\boldsymbol{Y}\\ &+ \frac{1}{6}(\tilde{u}_{112} - 2\tilde{u}_{121} + \tilde{u}_{211})\,ad^2_{\boldsymbol{X}}\boldsymbol{Y}\\ &+ \frac{1}{24}(-4\tilde{u}_{1121} + 4\tilde{u}_{1211})\,ad^3_{\boldsymbol{X}}\boldsymbol{Y}\\ &+ \frac{1}{120}(-4\tilde{u}_{11112} - 4\tilde{u}_{11121} + 16\tilde{u}_{11211}\\ &- 4\tilde{u}_{12111} - 4\tilde{u}_{21111})\,ad^4_{\boldsymbol{X}}\boldsymbol{Y} + \dots\end{aligned} \tag{8}$$

with the abbreviated notation

$$\tilde{u}_{i_1 i_2 \dots i_k} = \int\limits_{T_k(t)} u_{i_1}(s_1)u_{i_2}(s_2)\dots u_{i_k}(s_k)\,\mathrm{d}s^k. \tag{9}$$

Now, for controls with more flexibility in phases that slightly extend the classical form [9]

$$u_1(t) = a_1\sin(k_1 t + \phi_1), \quad u_2(t) = a_2\sin(k_1 t + \phi_2) \tag{10}$$

their parameters a_1, a_2, k, ϕ_1, ϕ_2, will be determined to generate the energy optimal vector field $ad^1_{\boldsymbol{X}}\boldsymbol{Y}$ as the main part of the series (3). To simplify calculations, the time horizon is fixed and equal to $T = 2\pi$, similarly to [9]). It is worth noticing that coefficients of generators in equation (8) vanish for controls (10) and the parameter k_1 is the same for both controls (otherwise coefficient of $ad^1_{\boldsymbol{X}}\boldsymbol{Y}$ zeroes) and parameters k_1, ϕ_1, ϕ_2 are to be optimized with respect to the energy performance index

$$\int\limits_0^T \left(u_1^2(t) + u_2^2(t)\right)\mathrm{d}t = \frac{T}{2}\left(a_1^2 + a_2^2\right). \tag{11}$$

Substituting controls (10) into equation (8), with a little assistance of computational algebra packages (like Mathematica), the following coefficient pre-multiplying $ad^1_{\boldsymbol{X}}\boldsymbol{Y}$ is obtained

$$\begin{aligned}&\frac{1}{2}\int\limits_0^{T=2\pi}\int\limits_0^{s_2}(u_1(s_1)u_2(s_2) - u_2(s_1)u_1(s_2))\,\mathrm{d}s_1\,\mathrm{d}s_2\\ &\qquad = \frac{a_1 a_2 \pi\sin(\phi_1 - \phi_2)}{k_1}.\end{aligned} \tag{12}$$

The maximum value of the coefficient is obtained for $k_1 = 1$, $\phi_1 - \phi_2 = \pi/2$ and $|a_1| = |a_2|$. Similar reasoning, but with much more computations, allows one to get expressions of coefficients multiplying $ad^r_{\boldsymbol{X}}\boldsymbol{Y}$, $r = 2,3,4$ with the frequency of the harmonic term in control u_2 increased

$$u_1(t) = a_1\sin(k_1 t + \phi_1), \quad u_2(t) = a_2\sin(r\cdot k_1 t + \phi_2), \tag{13}$$

in the following form

$$\begin{cases} -\dfrac{a_1^2 a_2 \pi \sin(2\phi_1 - \phi_2)}{4k_1^2} & \text{for } r = 2, \\ \dfrac{a_1^3 a_2 \pi \sin(3\phi_1 - \phi_2)}{24k_1^3} & \text{for } r = 3, \\ -\dfrac{a_1^4 a_2 \pi \sin(4\phi_1 - \phi_2)}{192k_1^4} & \text{for } r = 4. \end{cases} \tag{14}$$

Based on (12), (14), a recursive expression can be guessed for the coefficient of $ad_{\boldsymbol{X}}^r \boldsymbol{Y}$ corresponding to the vector field in the $(r+1)$-st layer of the Ph. Hall basis

$$\frac{(-1)^{r+1} a_1^r a_2 \pi \sin(r\phi_1 - \phi_2)}{2^{r-1} r! k_1^r}, \qquad r = 1, \ldots. \tag{15}$$

with parameters maximizing (15) equal to

$$k_1 = 1, \qquad r\phi_1 - \phi_2 = \frac{\pi}{2}. \tag{16}$$

Now, amplitudes of controls a_1, a_2 are to be optimized assuming a fixed value of (15) equal to $\tilde{\Delta}$ and Conditions (16) satisfied. The standard Lagrange multiplier technique [29] was applied to minimize (11) subject to equality constraints in a general form

$$a_1^r a_2 - \Delta = 0, \qquad r = 1, 2, \ldots, \qquad \Delta = \frac{2^{r-1} r!}{(-1)^{r+1} \pi} \cdot \tilde{\Delta}, \tag{17}$$

For the Lagrange function, with a multiplier λ,

$$L(a_1, a_2, \lambda) = a_1^2 + a_2^2 - \lambda (a_1^r a_2 - \Delta) \tag{18}$$

stationary conditions were formulated

$$\frac{\partial L(a_1, a_2, \lambda)}{\partial a_1} = \frac{\partial L(a_1, a_2, \lambda)}{\partial a_2} = \frac{\partial L(a_1, a_2, \lambda)}{\partial \lambda} = 0 \tag{19}$$

and the optimal relationship between amplitudes obtained

$$|a_1| = |a_2| \sqrt{r}. \tag{20}$$

Two possible sets of signs are subordinated to the rules:

$$\begin{array}{lll} \operatorname{sgn}(a_1) = \pm 1, \quad \operatorname{sgn}(a_2) = \operatorname{sgn}(\Delta) & \text{for} & \text{even } r, \\ \operatorname{sgn}(a_1) \cdot \operatorname{sgn}(a_2) = \operatorname{sgn}(\Delta) & \text{for} & \text{odd } r. \end{array} \tag{21}$$

Particular values of a_1, a_2 are derived from equations (20), (21). It is worth noticing that manipulating with signs of a_1, a_2 can be viewed as a modification of initial phases according to the trigonometric identity $a\sin(\alpha) = -a\sin(\alpha + \pi)$.

Based on the presented derivations, the following conclusions can be formulated concerning mainly optimal values of parameters that impact the coefficient multiplying $ad_{\boldsymbol{X}}^r \boldsymbol{Y}$:

- base frequency of u_1, u_2 controls equal to $k_1 = 1$,
- frequency of u_2 equal to r,
- phase shifts have to obey (16),
- amplitudes of controls should satisfy (20),
- the number of parameters to optimize is greater than the number of constraints imposed, thus extra optimization can be performed to optimize other criteria besides the energy of motion,
- controls used to steer the coordinate corresponding to the r-th layer vector field, also impact a motion within all consecutive layers.

After a straightforward integration of equation (6) for controls (13), the configuration shifts collected in Table 1 were obtained. Comparing equation (15) with the data obtained, one can notice that they are the same up to some signs (the multiplier -1 in Table 1 results from $(-1)^{r+1}$ multiplied by $(-1)^r$ due to $ad_{\boldsymbol{X}}^r \boldsymbol{Y}$, (7)). It appears that for chained systems (6) the local Lie-algebraic method works also globally.

Table 1
Steering the $(r+2)$-th coordinate to its desired value

r	$dq_{r+2}(2\pi) = q_{r+2}(2\pi) - q_{r+2}(0) =$
1	$-a_1 a_2 \pi \sin(\phi_1 - \phi_2)$
2	$-a_1^2 a_2 \pi \sin(2\phi_1 - \phi_2)/4$
3	$-a_1^3 a_2 \pi \sin(3\phi_1 - \phi_2))/24$

Sinusoidal controls are frequently used in Lie algebraic methods of motion planning due to simple integration rules of control-dependent data. The same is also valid for polynomial controls. A motion planner for systems (6) using polynomial controls is presented in [30].

3. ALGORITHMS

The presentation of algorithms aimed at solving a motion planning task for two-input, one-chained system begins with a basic algorithm (as in [9] with slightly modified notations to fit with those used throughout this paper) which is modified later on. In algorithms presented $\boldsymbol{\Delta}^k$ denotes a vector from a current initial configuration to the goal one at the k-th stage of a particular algorithm. In basic Algorithm 1 it is assumed that no extra information is available, thus amplitudes of controls u_1, u_2 are assumed to be equal. Detailed steps of Algorithm 1 are presented in Listing 1.

Modified versions of Algorithm 1 differ from the basic one with Step 2 only, thus this step is presented. Algorithms 2, 3 share frequencies of controls u_1, u_2 but differ in a distribution of amplitudes of controls, phase differences and initial values of phases. In Algorithms 2, 3 phase shifts (16) and amplitudes (20) are set optimally. Additionally, in Algorithm 3 initial phase ϕ_2 is optimized at each stage while in Algorithm 2 it remains fixed. From a numerical point of view, Algorithms 2 and 3 are extremely fast. The more demanding one, Algorithm 3, requires only $(n-3)$ one dimensional optimizations ($n = \dim \boldsymbol{q}$).

Analyzing Algorithm 2 and 3, it can be noticed that discontinuity of the control u_2 appears as the phase $\phi_2 \neq \phi_1$ (if $\phi_2 = \phi_1$ no configuration shift is generated after applying controls on the interval $[0, T]$. Consequently, a planning task cannot

be solved). Controls in planning algorithms play an auxiliary role as the true input for control algorithms (that follows the planning stage) is a desired trajectory which is at least continuous even for discontinuous controls that generate it. For some systems like free-flying robots powered with jet thrusters, discontinuous controls are typical. For other systems when discontinuity is an issue some standard techniques of penalizing discontinuity can be applied. For this particular planning task a kind of ϕ_2 phase modulation could also be tried (its initial value preserves continuity of the control u_2 while its final value is the optimal one). It is worth mentioning that many classical methods of motion planning generate discontinuous controls [7]. When a cost function describes an overall time minimization, bang-bang controls are typical.

In Algorithms 1–3 consecutive coordinates are steered to their desired values applying controls with frequency of the second control increased. An interesting question can be posed: why not steer them in different (permuted) order? The answer is a direct consequence of the gCBHD formula (8). When one wants to control the $(r+2)$-nd coordinate (motion along the vector field $ad^r_{\boldsymbol{X}}\boldsymbol{Y}$ thus also higher order vector fields are generated $ad^s_{\boldsymbol{X}}\boldsymbol{Y}$, $s>r$ and they modify remaining coordinates, $(r+3,\ldots,n)$. It is possible to steer any coordinate first, say the $(r+3)$-rd before the $(r+2)$-nd, but when the $(r+2)$-nd coordinate will be steered afterwards into its desired value it will disturb the proper value of the $(r+3)$-rd coordinate. Consequently, coordinates should be steered into their desired values in a strictly determined order.

Algorithm 1 (basic one)

Initial data: system (6), boundary configuration $\boldsymbol{q}_0$, $\boldsymbol{q}_f$, time horizon $T=2\pi$.

Step 1. Compute $\boldsymbol{\Delta}^0=\boldsymbol{q}_f-\boldsymbol{q}_0$. Steer coordinates q_1, q_2 to their desired values using any controls (for example, constant ones). With the controls, integrate remaining equations (6), starting with $\boldsymbol{q}_0$, to get a new initial configuration $\boldsymbol{q}_0^1$ for $t=T$. Set $\boldsymbol{\Delta}^1=\boldsymbol{q}_f-\boldsymbol{q}_0^1=(0,0,\Delta_3^1,\ldots,\Delta_n^1)$. Assign to variable *en* the energy expenditure on controls at this stage. Initialize iterator $r\leftarrow 1$.

Step 2. Steer q_{r+2} coordinate to its final state using controls

$$u_1(t)=\pm b\sin(t),\quad u_2(t)=\pm b\cos(rt), \tag{22}$$

with appropriate values of b (amplitudes and signs) in u_1,u_2 to compensate the error Δ^r_{r+2}. Add the energy spent on this stage to the total energy $en\leftarrow en+2\pi b^2$.

Step 3. Check the stop condition:
if $(r=n-2)$ then output the total energy *en* and complete the algorithm, otherwise progress with Step 4.

Step 4. Increase the iterator $r\leftarrow r+1$. For selected controls, integrate equations (6) to get a new value of $\boldsymbol{q}_0^r$, for $t=T$, and update

$$\boldsymbol{\Delta}^r=\boldsymbol{q}_f-\boldsymbol{q}_0^r=(0,0,\ldots,0,\Delta^r_{r+2},\ldots,\Delta^r_n)$$

(with $(r+1)$ leading zeroes). Go to Step 2.

Algorithm 2 (basic improved: amplitudes and relationship of phases optimized, phase ϕ_2 fixed)

Step 2. Steer q_r to its final state with controls

$$\begin{aligned} u_1(t)&=\pm\sqrt{r}\cdot b\sin\{t+\phi_1(=(\pi/2-\phi_2)/r)\},\\ u_2(t)&=\pm b\sin(rt+\phi_2)\end{aligned} \tag{23}$$

with ϕ_2 fixed and b (amplitudes and signs) in controls u_1, u_2 selected appropriately to compensate the error Δ^r_{r+2}. Add the energy spent on this stage $en\leftarrow en+2\pi b^2(r+1)/2$.

Step 2a. (optional) When $(r<n-2)$ (not the last iteration) integrate the $(r+3)$-rd equation of system (6), corresponding to coordinate q_{r+1}, initialized at $\boldsymbol{q}_0^r$ and with selected controls. Let the value $q_{r+1}(T)=\xi_1$. Repeat the same procedure with the other admissible set of signs of controls u_1,u_2, to get $q_{r+1}(T)=\xi_2$. Select those controls which result in a smaller value of $|\boldsymbol{q}_f^{r+1}-q_{r+1}(T)|$ where $\boldsymbol{q}_f^{r+1}$ denotes the final value of the $(r+1)$-st coordinate of $\boldsymbol{q}_f$.

Algorithm 3 (Algorithm 2 with phase ϕ_2 optimized)

Step 2. Steer q_r to its final state with controls

$$\begin{aligned} u_1(t)&=\pm\sqrt{r}\cdot b\sin\{t+\phi_1(=(\pi/2-\phi_2)/r)\},\\ u_2(t)&=\pm b\sin(rt+\phi_2)\end{aligned} \tag{24}$$

with ϕ_2 fixed and b (amplitudes and signs) in controls u_1, u_2 selected appropriately to compensate the error Δ^r_{r+2}. Add the energy spent on this stage $en\leftarrow en+2\pi b^2(r+1)/2$.
When $(r<n-2)$ (not the last iteration) integrate the $(r+3)$-rd equation of system (6), corresponding to coordinate q_{r+1}, initialized at $\boldsymbol{q}_0^r$ and controls (24) with varied ϕ_2 to get $q_{r+1}(T)=\xi(\phi_2)$. Calculate the optimal value of $\phi_2^\star$

$$|\boldsymbol{q}_f^{r+1}-\xi(\phi_2^\star)|=\min_{\phi_2\in[0,2\pi]}|\boldsymbol{q}_f^{r+1}-\xi(\phi_2)|. \tag{25}$$

Progress later using controls (24) with $\phi_2=\phi_2^\star$.

4. SIMULATIONS

In all tests a motion between the initial configuration $\boldsymbol{q}_0=(0,0,0,0,0)^T$ and the goal one $\boldsymbol{q}_f=(0,0,-4,4,4)^T$ is planned and phases are expressed in grads [°].

In tables configurations obtained after each iteration of the tested three algorithms are presented together with data uniquely determining controls. A five-dimensional system (6) is selected because the first iteration, $r=1$, does not properly differ amplitudes of controls (in equations (22), (23), (24) amplitudes are the same as $\sqrt{r}=1$), while the last one does not affect the energy expenditure on controls at all (there is no next coordinate to optimize), thus $n=5$ is the minimal dimension to notice any difference. In the first simulation, Table 2, basic Algorithm 1 was run and in each iteration either a random of two possible solutions (21) was selected (top panel) or the better one (bottom panel). As expected, the latter selection generates energetically better solution; however, it requires a little bit more computational effort than the former one.

Table 2
Basic Algorithm 1: without optimization (i.e. the first admissible set of two possible values of u_1, u_2 is selected); with optimization among the two; $\phi_1 = 0$, $\phi_1 = 90^\circ$ are fixed in both cases

r	$\boldsymbol{q}_0^r$	a_1	a_2	en_r
	without			
0	$(0,0,0,0,0)$	+1.13	-1.13	8.00
1	$(0,0,-4.,4.51,-3.18)$	-0.87	-0.87	4.73
2	$(0,0,-4,4,-3.63)$	+2.76	+2.76	47.97
	$(0,0,-4,4,4)$	$\sum_r en_r = 60.70$		
	with			
0	$(0,0,0,0,0)$	+1.13	-1.13	8.00
1	$(0,0,-4.,4.51,-3.18)$	+0.87	-0.87	4.73
2	$(0,0,-4,4,-2.74)$	+2.68	+2.68	45.08
	$(0,0,-4,4,4)$	$\sum_r en_r = 57.81$		

In Table 3 simulation results were collected, for amplitudes (20) and the relationship of phases (16) optimized but with the phase ϕ_2 fixed. It appears that the initial phase shift impacts energy expenditure considerably. Results of running Algorithm 3 were presented in Table 4. The algorithm is the most computationally involved as it requires one dimensional optimization in each iteration. It may look strange that, for this particular set of data, it generated even worse results than Algorithm 2, cf. Table 3 (the bottom panel). A more careful look at configurations $\boldsymbol{q}_0^2$ reveals that the fourth component of $\boldsymbol{q}$ generated with Algorithm 3 ($= 4.24$) is closer to the $\boldsymbol{q}_f^4 = 4$ than that generated with Algorithm 2 ($= 2.26$). It can be concluded that the presented algorithms are optimal locally and even the best local decisions do not necessarily generate the globally best solution.

A selection between two possible solutions (cf. equation (21)) is tested once again, this time when the best and the worst selection is made in each iteration. Results are presented

Table 3
Algorithm 2: amplitudes optimized, phases coupled optimally, but without optimization of an initial phase, ϕ_2 fixed

r	$\boldsymbol{q}_0^r$	a_1	a_2	en_r
	$\phi_2 = 90^\circ$			
0	$(0,0,0,0,0)$	-1.13	-1.13	8.00
1	$(0,0,-4.,4.51,-3.18)$	-0.97	+0.69	4.47
2	$(0,0,-4,4,-3.18)$	+3.12	-1.80	40.84
	$(0,0,-4,4,4)$	$\sum_r en_r = 53.31$		
	$\phi_2 = 30^\circ$			
0	$(0,0,0,0,0)$	-1.13	-1.13	8.00
1	$(0,0,-4.,2.26,-1.27)$	-1.46	-1.04	10.10
2	$(0,0,-4,4,0)$	+2.70	-1.56	30.46
	$(0,0,-4,4,4)$	$\sum_r en_r = 48.57$		

Table 4
Results of running Algorithm 3

r	$\boldsymbol{q}_0^r$	ϕ_1	ϕ_2	a_1	a_2	en_r
0	$(0,0,0,0,0)$	200	110	-1.13	-1.13	8.00
1	$(0,0,-4.,4.24,-2.89)$	180	270	-0.76	+0.53	2.70
2	$(0,0,-4,4,-2.70)$	0	90	+3.06	-1.77	39.45
	$(0,0,-4,4,4)$	$\sum_r en_r = 50.15$				

in Table 5. It appears that the worst selections can bring a really energy ineffective motion. In the last test, Table 6, it was illustrated how badly selected and fixed phases (not satisfying Rule (16)) impact the energy of motion. A substantial increase of energy is clearly visible in this case.

Table 5
The amplitude optimization, phases fixed $\phi_1 = 0^\circ$ $\phi_2 = 90^\circ$. The selected solution: either the best or the worst of the two

r	$\boldsymbol{q}_0^r$	a_1	a_2	en_r
	the best			
0	$(0,0,0,0,0)$	+1.13	-1.13	8.00
1	$(0,0,-4.,4.51,-3.18)$	+0.97	-0.69	4.47
2	$(0,0,-4,4,-2.68)$	+3.07	1.77	39.39
	$(0,0,-4,4,4)$	$\sum_r en_r = 51.86$		
	the worst			
0	$(0,0,0,0,0)$	-1.13	+1.13	8.00
1	$(0,0,-4.,-4.51,-3.18)$	+2.48	+1.76	29.08
2	$(0,0,-4,4,-24.33)$	+4.40	+2.54	81.10
	$(0,0,-4,4,4)$	$\sum_r en_r = 118.18$		

Table 6
Algorithm 2: fixed phases $\phi_1 = 60^\circ$, $\phi_2 = 30^\circ$ selected badly, i.e. the last rule in (16) is not satisfied

r	$\boldsymbol{q}_0^r$	a_1	a_2	en_r
0	$(0,0,0,0,0)$	+1.60	1.60	16.00
1	$(0,0,-4.,3.19,-2.55)$	-1.13	-0.80	6.05
2	$(0,0,-4,4,-2.09)$	+3.56	-2.06	53.17
	$(0,0,-4,4,4)$	$\sum_r en_r = 75.22$		

5. CONCLUSIONS

In this paper the problem of optimizing energy of motion for two-input one-chained systems was stated and solved. Based on the very general Campbell-Baker-Hausdorff-Dynkin formula, applied to the systems, some equations (constraints) were formulated relating amplitudes and phases of controls. Consequently, a dimension of a search space for a locally optimal solution was decreased. It appeared that it is desirable to opti-

mize an energy of motion not only by setting a relationship between amplitudes and phases optimally for a currently steered coordinate of a configuration vector but also to use some kind of prediction of trajectories of a consecutive coordinate. Two new algorithms were designed and their outputs compared with the classical one known from the robotic literature. About 20% decrease in the energy expenditure on controls was observed using the optimized algorithms. The algorithms are designed for this special class of systems thus they are extremely fast and their complexity increases linearly with the configuration space dimension. Moreover, neither numerical nor singularity problems are encountered. An interesting fact was also discovered that for this very special chained-form systems with only one vector field in each layer (instead of many more predicted by general considerations) the local Lie algebraic method works also globally.

In the nearest future, we are planning to extend the presented method to multi-chained systems and to propose a new cost function based not only on a predicted evolution of the next coordinate but also on all coordinates not properly steered yet.

REFERENCES

[1] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[2] M. Galicki, "The planning of optimal motions of non-holonomic systems," *Nonlinear Dyn.*, vol. 90, pp. 2163–2184, 2017.

[3] T. Rybus, "The obstacle vector field (ovf) method for collision-free trajectory planning of free-floating space manipulator," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 70, no. 2, p. e140691, 2022, doi: 10.24425/bpasts.2022.140691.

[4] W. Wawrzyński, M. Zieja, M. Żokowski, and N. Sigiel, "Optimization of autonomous underwater vehicle mission planning process," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 70, no. 2, p. e140371, 2022, doi: 10.24425/bpasts.2022.140371.

[5] I. Duleba, W. Khefifi, and I. Karcz-Duleba, "Layer, Lie algebraic method of motion planning for nonholonomic systems," *J. Frankl. Inst.*, vol. 349, no. 1, pp. 201–215, 2012.

[6] A. Ratajczak and K. Tchoń, "Parametric and non-parametric jacobian motion planning for non -holonomic robotic systems," *J. Intell. Robot. Syst.*, vol. 77, no. 3, pp. 445–456, 2015.

[7] L. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *Am. J. Math.*, vol. 79, no. 3, p. 497–516, 1957, doi: 10.2307/237256.

[8] S. Agraval and S. Bhattacharya, "Optimal control of driftless systems: Some new results," in *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1999, pp. 356–261.

[9] R. Murray and S. Sastry, "Steering nonholonomic control systems using sinusoids," *IEEE Trans. Autom. Control*, vol. 38, no. 5, pp. 700–716, 1993.

[10] D.E. Kirk, *Optimal Control Theory: An Introduction*. Prentice-Hall, 1970.

[11] S.L. Richter and R. DeCarlo, "Continuation methods: Theory and applications," *IEEE Trans. Circuits Syst.*, vol. 30, no. 6, pp. 347–352, 1983.

[12] S. LaValle and J. Kuffner, *Algorithmic and Computational Robotics: New Directions*. A.K. Peters/CRC Press, 2001, ch. Rapidly-exploring random trees: Progress and prospects, pp. 293–308.

[13] R. Geraerts and M.H. Overmars, *Algorithmic Foundations of Robotics*. Springer Berlin Heidelberg, 2004, ch. A Comparative Study of Probabilistic Roadmap Planners, pp. 43–57, doi: 10.1007/978-3-540-45058-0_4.

[14] R. Brockett, "Control theory and singular Riemannian geometry," in *New Directions in Applied Mathematics*. New York: Springer-Verlag, 1981, pp. 11–27.

[15] C. Altafini, "The general n-trailer problem: Conversion into chained form," in *Conf. on Decision and Control*, Tampa, 1998, pp. 3129–3130.

[16] W. Kowalczyk and K. Kozłowski, "Trajectory tracking and collision avoidance for the formation of two-wheeled mobile robots," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 67, no. 5, pp. 915–924, 2019.

[17] M. Ito, "Motion planning of a second-order nonholonomic chained form system based on holonomy extraction," *Electronics*, vol. 8, no. 11, pp. 1–21, 2019.

[18] J. Luo and P. Tsiotras, "Control design for chained-form systems with bounded inputs," *Syst. Control Lett.*, vol. 39, no. 2, pp. 123–131, 2000.

[19] Z.-P. Jiang and H. Nijmeijer, "A recursive technique for tracking control of nonholonomic systems in chained form," *IEEE Trans. Autom. Control*, vol. 44, no. 2, pp. 265–279, 1999.

[20] R. Murray and S. Sastry, "Steering nonholonomic systems in chained form," in *Proc. Conf. on Decision and Control*, Brighton, 1991, pp. 2121–2126.

[21] D. Tilbury, J. Laumond, R. Murray, S. Sastry, and G. Walsh, "Steering car-like systems using sinusoids," in *IEEE Conf. on Robotics and Automation*, Nice, France, 1992, pp. 1993–1997.

[22] F. Jean, *Control of Nonholonomic Systems: from Sub-Riemannian Geometry to Motion Planning*, ser. Springer Briefs in Mathematics. Springer, 2014.

[23] B. Jakubczyk, "Nonholonomic path following with fastly oscillating controls," in *IEEE Int. Workshop on Robot Motion and Control*, Wasowo, Poland, 2013, pp. 99–103.

[24] I. Duleba, "Impact of control representations on efficiency of local nonholonomic motion planning," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 59, no. 2, pp. 213–218, 2011.

[25] J. Laumond, S. Sekhavat, and F. Lamiraux, *Robot Motion Planning and Control*, ser. Lecture Notes in Control and Information Sciences. Springer Berlin, Heidelberg, 1998, vol. 229, ch. Guidelines in Nonholonomic Motion Planning for Mobile Robots.

[26] R. Strichartz, "The Campbell-Baker-Hausdorff-Dynkin formula and solutions of differential equations," *J. Funct. Anal.*, vol. 72, pp. 320–345, 1987.

[27] W. Chow, "Über Systeme von linearen partiellen Differentialgleichungen erster Ordnung," *Math. Ann.*, vol. 117, no. 1, pp. 98–105, 1939.

[28] I. Duleba and W. Khefifi, "Pre-control from of the gcbhd formula and its application to nonholonomic motion planning," in *Proc. 7th IFAC Symposium on Robot Control*, vol. 1. Wrocław: Oxford, Elsevier, 2004, pp. 123–128.

[29] D. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Belmont, Massachusetts: Athena Scientific, 1996.

[30] D. Tilbury, R. Murray, and S. Sastry, "Trajectory generation for the n-trailer problem using Goursat normal form," *IEEE Trans. Autom. Control*, vol. 40, no. 5, p. 802–819, 1995.