Robotica (2016) volume 34, pp. 2223–2240.
doi:10.1017/S0263574714002859

Motion planning of strongly controllable stratified systems

Ignacy Duleba* and Michal Opalka

Chair of Cybernetics and Robotics, Electronics Faculty, Wroclaw University of Technology, Janiszewski St. 11/17, 50-372 Wroclaw, Poland

(Accepted December 3, 2014. First published online: January 16, 2015)

SUMMARY

In this paper a motion planner for nonholonomic stratified systems was proposed. Those systems may arise easily when reliable systems are designed to be robust against failures in difficult servicing environments. For a special class of the systems, a strong controllability condition was introduced, and a criterion to satisfy the condition was formulated and used to plan the motion of free-floating space manipulators. Modules of the planner were enumerated and their roles were emphasized. Some features of the planner were examined and discussed based on simulation results performed on two models of space manipulators.

KEYWORDS: Motion planning; Stratified systems; Nonholonomic systems; Space manipulators.

1. Introduction

Motion planning is one of the classical tasks of robotics. It relies on planning the motion of a robot (an agent) between two boundary configurations, satisfying some restrictions on available controls and the robot's surrounding (obstacle avoidance) and possibly optimizing a quality function. Many methods have been developed to solve the task.[1] They differ in models accepted (continuous, discrete), their special features (nilpotent, in a chain form, flat) or in environmental description (obstacles in a form of polygons, spheres, ellipsoids, or environments with special structures like labyrinths). Some time ago, only objects described by a single model were considered. Recently, a lot of attention is paid to hybrid systems that couple models from different domains (continuous–discrete to model production processes) or to different models within a single domain (for example switched systems or a legged locomotion). Stratified systems, being the main interest of this paper, are a special subclass of the second type of aforementioned hybrid systems.[2,3] They are characterized by a global configuration vector that is a union of configuration vectors of a few subsystems. For some reasons, the system cannot be controlled within the global configuration space. However, while purposefully switching on and off its subsystems (only one can be active at each time point) a desired state of the global configuration is obtained. Switching between subsystems is triggered either at particular states (like in walking robots, when one leg touches the ground, the other can move it up) or when a switch can be performed at any time. The latter models will be considered in this paper and their practical importance originates from reliability theory. It can be observed that at each time point only some joints of the manipulator can be directly controlled and that a total configuration space has got more dimensions than the number of controls. A stratified system perfectly models the situation when a failure of one or more motors of a space manipulator can be compensated with a control strategy using a gear box system designed to transmit momentum to all links from motors in working order.

Methods to steer stratified systems should be based on classical methods of controlling single model systems as the theory for such systems is well developed. A formal mathematical background for stratified nonholonomic systems was laid by Goodwine.[2] His ideas were extended and translated into algorithms by Harmati.[3,4] However, none of the authors have considered phenomena arising

* Corresponding author. E-mail: ignacy.duleba@pwr.edu.pl

while switching (for example how to preserve continuity of controls). To fill the gap, at least for a special sub-class of stratified systems introduced in this paper and called strongly controllable stratified systems, a motion planning algorithm was designed to plan the motion of stratified systems.

The presented approach belongs to a class of analytical and continuous methods of motion planning. Usually the methods generate final results fast and with some desirable properties (like continuity) but have got some problems with inadmissible areas in the state space. In this case, discrete methods (RRT and other methods[1]) can support analytical methods to detect obstacle-free areas. On the other hand, analytical methods can support discrete methods in smoothing paths (trajectories) generated with the use of discrete methods or in the optimization of their shapes.

This paper is organized as follows. In Section 2, the basic terminology and definition of (nonholonomic) stratified systems is presented and the task of stratified motion planning is defined. In this section, strongly controllable stratified systems are introduced and a controllability condition is defined to check the property. From a practical point of view, strongly controllable stratified systems admit a nice decomposition of a global motion planning task into a sequence of subtasks for subsystems. In Section 3, two models of nonholonomic stratified systems are presented.

Two general-purpose methods of nonholonomic motion planning will be recalled in Section 4. The Newton method and the Lie-algebraic method will be used as a tool for solving sub-planning tasks. The first method is well suited to plan long-range motions in obstacle-free environments while the second one plans short-range motion in an obstacle cluttered space. With the basic components of a motion planner explained, a solution of a sequence of tasks, with a proper concatenation of controls, will be presented. Hybridization of planning methods will be possible, i.e. each of the sub-plannings may be performed by a different algorithm, and the results could still be interpreted as a single control scenario. In Section 5, a complete algorithm of nonholonomic motion planning of stratified systems is given. Its main components are discussed and their roles in a motion planning task are highlighted. Simulation results performed on two models of underactuated space robots are collected in Section 6. Section 7 concludes the paper.

2. Stratified Systems

For stratified systems,[2] a global (unconstrained) system $\mathcal{S}_0$ is composed of a few (constrained) subsystems (strata), $\mathcal{S}_1, \ldots, \mathcal{S}_i$. Each of them is continuous (nonlinear, and later nonholonomic) and results from setting constraints on its parent subsystem. Physically, adding constraints corresponds to switching between a parent and its offspring system. This way subsystems are connected with others by a graph of possible subsystem switching. A configuration space of a descendant (more constrained than its parent) subsystem inherits some part of the configuration space of its parent together with its truncated (projected) equations. By definition, a motion within the global unconstrained system $\mathcal{S}_0$ is not allowed. The formal definition of (nonholonomic) stratified systems follows:[2]

For a stratified configuration manifold $\mathcal{M}$, consisting of strata $\{\mathcal{S}_0, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_{12} \ldots, \mathcal{S}_i\}$, where multi-index k collects the numbers of all additional constraints imposed on stratum $\mathcal{S}_k$, the nonholonomic stratified system is defined by a set of equations:

$$\begin{array}{lll}
\mathcal{S}_0 : \dot{\boldsymbol{q}}|_{\mathbb{T}\mathcal{S}_0} = \boldsymbol{g}^0_1(\boldsymbol{q}|_{\mathcal{S}_0}) \cdot u^0_1 + \cdots + \boldsymbol{g}^0_{m_0}(\boldsymbol{q}|_{\mathcal{S}_0}) \cdot u^0_{m_0} & \longrightarrow & \Sigma_0 \\
\mathcal{S}_1 : \dot{\boldsymbol{q}}|_{\mathbb{T}\mathcal{S}_1} = \boldsymbol{g}^1_1(\boldsymbol{q}|_{\mathcal{S}_1}) \cdot u^1_1 + \cdots + \boldsymbol{g}^1_{m_1}(\boldsymbol{q}|_{\mathcal{S}_1}) \cdot u^1_{m_1} & \longrightarrow & \Sigma_1 \\
\qquad\vdots & & \\
\mathcal{S}_i : \dot{\boldsymbol{q}}|_{\mathbb{T}\mathcal{S}_i} = \boldsymbol{g}^i_1(\boldsymbol{q}|_{\mathcal{S}_i}) \cdot u^i_1 + \cdots + \boldsymbol{g}^i_{m_i}(\boldsymbol{q}|_{\mathcal{S}_i}) \cdot u^i_{m_i} & \longrightarrow & \Sigma_i,
\end{array} \tag{1}$$

where operation $(\boldsymbol{q}|_{\mathcal{S}_k})$ projects (truncates) configuration $\boldsymbol{q}$ from manifold $\mathcal{M}$ into sub-manifold (stratum) $\mathcal{S}_k$.

The projection is defined not only for the configuration but also for the velocity space. To distinguish one from another, notation $\mathbb{T}\mathcal{S}$ was used for the velocity space. A dimensionality of the global configuration vector $\boldsymbol{q}|_{\mathcal{S}_0}$ is equal to $\dim \mathcal{S}_0 = n_0$ and it is bigger than the dimensionality of configuration vectors of subsystems $\Sigma_i(\mathcal{S}_i)$, $n_0 > n_i = \dim \mathcal{S}_i$. Moreover, each subsystem is nonholonomic, which implies that $n_i > m_i$. Each subsystem Σ_k may also have different numbers

https://doi.org/10.1017/S0263574714002859 Published online by Cambridge University Press

of inputs. Some controls can be shared between subsystems, i.e. the lth input u_l^k of subsystem Σ_k may represent the same physical input as the jth control u_j^z of subsystem Σ_z. Similarly, motions within certain subsystems may modify the same coordinate(s). That is, $\boldsymbol{q}|_{\mathcal{S}_l}$ may share one or more coordinates with $\boldsymbol{q}|_{\mathcal{S}_k}$.

Now, a stratified motion planning can be defined:

For a given stratified system in Eq. (1) *determine a sequence of strata* $\mathcal{S}_I, \ldots \mathcal{S}_{mid}, \ldots, \mathcal{S}_D$ *and corresponding sequence of open-loop controls* $\boldsymbol{u}^I(\cdot), \ldots, \boldsymbol{u}^{mid}(\cdot), \ldots, \boldsymbol{u}^D(\cdot) \in \mathbb{L}_m^2[0, T_i]$, $t \in [0, T_i]$, *steering the system from a given initial* $\boldsymbol{q}_0 \in \mathcal{S}_0$ *to the goal state* $\boldsymbol{q}_d \in \mathcal{S}_0$,

where $\mathbb{L}_m^2[0, T_i]$ is a space of m-copies of square integrable functions defined on interval $[0, T_i]$ and T_i denotes a time horizon (selected by a user) of the ith planning. It is assumed that the global time-dependent characteristics (controls, trajectory) are retrieved based on pieces obtained from sub-planning and, shifted on the time axis appropriately (for simplicity, each planning starts at zero on the time axis). A time interval for each sub-task is selected by a user. In general, solving a basic stratified motion planning task requires admissible controls that steer a system from its initial configuration to the desired one to be found. Boundary configurations of the motion planning tasks are defined on the highest (unconstrained) stratum $\mathcal{S}_0$, but motion on it is inadmissible.

Motion planning of stratified systems is difficult, because not only a method of solving a planning task for a selected subsystem should be known, but also a switching scenario between permissible subsystems is to be designed. Each subsystem is defined on a different manifold, thus initial and final configurations for planning the sub-task may be defined in different manifolds (likely with different dimensionality) as well. However, results of the planning among subsystems coupled properly should form a control scenario that solves an imposed motion planning task. As some coordinates can be shared between subsystems, a planning for a current subsystem should not disturb desirable effects obtained from previous plannings. Additionally (un-)shared inputs are also important in motion planning. If one needs to obtain smooth controls, the planning algorithm should zero values of unused controls and those ones shared between strata should be, at least, continuous.

A primary question in motion planning is whether the system is controllable, i.e. that for any pair of boundary configurations whether there exist controls solving the task. For stratified systems in Eq. (1), Goodwine[2] adapted classical concepts known for smooth systems and formulated the stratified controllability

Given a stratified configuration manifold and a collection of strata, $\{\mathcal{S}_{i_1}, \mathcal{S}_{i_2}, \ldots, \mathcal{S}_{i_m}\}$*, a system is small time locally stratified controllable if reachable set* $\mathbb{R}_T(\boldsymbol{q}_0)$ *at time* T *initialized at configuration* $\boldsymbol{q}_0$ *contains a neighborhood of* $\boldsymbol{q}_0$ *in* $\mathcal{S}_{i_1} \cup \mathcal{S}_{i_2} \cup \ldots \cup \mathcal{S}_{i_m}$*, for all* $\boldsymbol{q}_0 \in \mathcal{S}_{i_1} \cup \mathcal{S}_{i_2} \cup \ldots \cup \mathcal{S}_{i_m}$ *and small enough* $T > 0$.

The controllability criterion for stratified systems exploits the geometric properties of the system to discover its motion capabilities. Below, one test for controllability is presented (this one and others can be found in ref. [2]).

Theorem 1. Let $\mathbb{T}_{\boldsymbol{q}_0}\mathcal{M}$ *be the tangent space of* $\mathcal{M}$ *at* $\boldsymbol{q}_0$ *and let* $\bar{\Delta}_{\mathcal{S}_i}|_{\boldsymbol{q}_0}$ *denote an involute closure of a distribution spanned by the vector fields of stratum* $\mathcal{S}_i$ *at* $\boldsymbol{q}_0$*. If there exists a nested sequence of strata*

$$\boldsymbol{q}_0 \in \mathcal{S}_p \subset \mathcal{S}_{p-1} \subset \ldots \subset \mathcal{S}_1 \subset \mathcal{S}_0,$$

such that the involute closures of distributions (of strata) fulfill the condition

$$\sum_{i=0}^{p} \bar{\Delta}_{\mathcal{S}_i}|_{\boldsymbol{q}_0} = \mathbb{T}_{\boldsymbol{q}_0}\mathcal{M}, \tag{2}$$

then the system is locally stratified controllable at $\boldsymbol{q}_0$.

One should notice that the criterion in Eq. (2) does not require full controllability on each stratum (subsystem). Consider a biped walking robot described by equations (this model is equivalent to a rowboat[4] or hexapod robot[2])

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi}_1 \\ \dot{\phi}_2 \end{pmatrix} = \begin{bmatrix} \alpha_1 \cos(\theta) & \alpha_2 \cos(\theta) \\ \alpha_1 \sin(\theta) & \alpha_2 \sin(\theta) \\ \alpha_1 & -\alpha_2 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \tag{3}$$

where components of the configuration vector are the following: x, y, and θ are the position and orientation of the robot, while ϕ_1, ϕ_2 are (directly controlled) the angles of legs with respect to the body. Functions α_1, α_2, assigned to each leg, take either the value of zero when the ith leg is in contact with the ground or a value of one if not in contact with the ground. Four strata can be defined for the robot, $\mathcal{S}_0$ ($\alpha_1 = \alpha_2 = 1$, inadmissible one), $\mathcal{S}_1$ ($\alpha_1 = 1, \alpha_2 = 0$, when the left leg is on the ground, coordinate ϕ_2 is changed freely, without disturbing other coordinates), $\mathcal{S}_2$ ($\alpha_1 = 0, \alpha_2 = 1$, when the right leg is on the ground and ϕ_1 free to move), $\mathcal{S}_{12}$ ($\alpha_1 = \alpha_2 = 0$, when both legs are on the ground, modifications of any ϕ_i will influence coordinates (x, y, θ). It is obvious that system in Eq. (3) is not controllable on each stratum $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_{12}$ but, as everyday experience shows, is globally controllable with condition in Eq. (2) satisfied.

For systems considered in this paper we require a stronger version of stratified controllability given as

$$\forall_i \in \{1, \ldots\} \quad \dim(\bar{\Delta}_{\mathcal{S}_i}) = \dim(\mathcal{S}_i) = n_i. \tag{4}$$

Condition (4) describes a small time local controllability[5] (STLC) on each stratum and implies controllability in Eq. (2) (the reverse implication does not hold). Systems satisfying Eq. (4) will be called *strongly controllable stratified systems*. STLC assures that not only controllability on each stratum is preserved but also maneuvers to reach local (short-distance) goals can be performed with only small modifications of a configuration vector. This feature is especially important to plan motion in obstacle cluttered environments when long-range motion can be decomposed into a series of sub-tasks with sub-goals (consecutive goals placed not too far from each other) selected to avoid obstacles. STLC is commonly assumed by Lie-algebraic methods.

3. Models of Underactuated Space Manipulators

Free floating manipulators are nonholonomic systems. Their nonholonomic constraints result from an angular momentum conservation law.[6] Although it is possible to make a space manipulator holonomic (supplying it with momentum wheels,[7,8] or even free-flying using thruster-jets[9]), it is undesirable for energy reasons. In the low-orbit environment jet propulsion fuel is non-renewable and electrical energy for momentum wheels is expensive. Moreover, the total mass of the spacecraft directly translates into mission costs. Either to reduce the total mass or to act reliably in emergency situations when one or more motors are damaged, a practical problem arises when a small number of physical motors are to be multiplexed to steer a large number of joints. Those models are naturally described as nonholonomic stratified systems. Two of them will be presented below.

The first model is a one-arm free-floating space manipulator, Fig. 1(a), composed of three-joints. Instead of presenting its analytical (two-pages long) model, some numeric values were chosen to simplify matters: $m_1 = m_2 = m_3 = 1$, $m_0 = 10$, $l_1 = l_2 = l_3 = 1$, $a = 1$, $b = 0.6$. The numerical values are rather unrealistic, as a base is usually much heavier than the links of the manipulator. However, such data impacts the dynamics of the system heavily and makes simulations more vivid. A driftless nonholonomic system derived from an angular conservation law is described by equations defined on stratum $\mathcal{S}_0$

$$\dot{\boldsymbol{q}} = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{\theta} \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ A_1(\boldsymbol{q}) & A_2(\boldsymbol{q}) & A_3(\boldsymbol{q}) \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \sum_{i=1}^{3} \boldsymbol{g}_i(\boldsymbol{q}) u_i = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{u}, \tag{5}$$

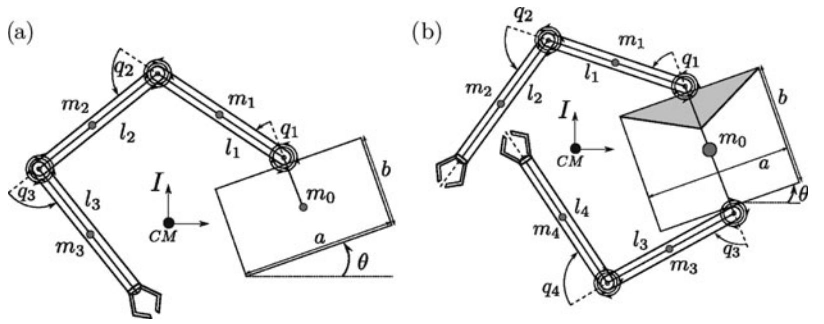

Fig. 1. (a) One-arm, (b) two-arm space manipulators in their inertial frames I originated at the center of mass (CM).

where

$$
\begin{aligned}
A_i(\boldsymbol{q}) &= f_i(\boldsymbol{q})/f_\theta(\boldsymbol{q}), \quad i = 1, 2, 3 \\
f_1(\boldsymbol{q}) &= -(90\, c_1 + 378\, c_2 + 138\, c_3 + 54\, c_{12} + 126\, c_{23} + 18\, c_{123} + 636) \\
f_2(\boldsymbol{q}) &= -(189\, c_2 + 138\, c_3 + 54\, c_{12} + 63\, c_{23} + 18\, c_{123} + 308) \\
f_3(\boldsymbol{q}) &= -(69\, c_3 + 63\, c_{23} + 18\, c_{123} + 88) \\
f_\theta(\boldsymbol{q}) &= 180\, c_1 + 378\, c_2 + 138\, c_3 + 108\, c_{12} + 126\, c_{23} + 36\, c_{123} + 845.2.
\end{aligned}
\tag{6}
$$

In Eqs. (6) and (11), a standard robotic convention to denote the sine/cosine function of angles was utilized, i.e. $c_{23} = \cos(q_2 + q_3)$, $s_1 = \sin(q_1)$. A stratification of the model is based on the assumption that two joints can be actuated at the same time while the remaining one is fixed. Therefore, motion planning is to be performed using lower strata $\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$, corresponding to fixed joints q_1, q_2, q_3, respectively.

From system in Eq. (5) its subsystems (on each stratum) are easy to derive. Every sub-stratum has got a dimension of one smaller than the configuration space of the fully actuated system and the projection $(\cdot)|_{\mathcal{S}_i}$, $i = 1, 2, 3$, will omit one, currently fixed, coordinate. Similarly, generators $\boldsymbol{g}_k$ will be appropriately truncated to get generators $\boldsymbol{g}_k^i$. Equations on the strata follow

$$
\mathcal{S}_1 : \qquad \dot{\boldsymbol{q}}|_{\mathbb{T}\mathcal{S}_1} = \begin{pmatrix} \dot q_2 \\ \dot q_3 \\ \dot\theta \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ A_2(\boldsymbol{q}) & A_3(\boldsymbol{q}) \end{bmatrix} \begin{pmatrix} u_2 \\ u_3 \end{pmatrix} = \boldsymbol{g}_2^1(\boldsymbol{q}|_{\mathcal{S}_1})u_2 + \boldsymbol{g}_3^1(\boldsymbol{q}|_{\mathcal{S}_1})u_3, \tag{7}
$$

$$
\mathcal{S}_2 : \qquad \dot{\boldsymbol{q}}|_{\mathbb{T}\mathcal{S}_2} = \begin{pmatrix} \dot q_1 \\ \dot q_3 \\ \dot\theta \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ A_1(\boldsymbol{q}) & A_3(\boldsymbol{q}) \end{bmatrix} \begin{pmatrix} u_1 \\ u_3 \end{pmatrix} = \boldsymbol{g}_1^2(\boldsymbol{q}|_{\mathcal{S}_2})u_1 + \boldsymbol{g}_3^2(\boldsymbol{q}|_{\mathcal{S}_2})u_3, \tag{8}
$$

$$
\mathcal{S}_3 : \qquad \dot{\boldsymbol{q}}|_{\mathbb{T}\mathcal{S}_3} = \begin{pmatrix} \dot q_1 \\ \dot q_2 \\ \dot\theta \end{pmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ A_1(\boldsymbol{q}) & A_2(\boldsymbol{q}) \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \boldsymbol{g}_1^3(\boldsymbol{q}|_{\mathcal{S}_3})u_1 + \boldsymbol{g}_2^3(\boldsymbol{q}|_{\mathcal{S}_3})u_2. \tag{9}
$$

Each subsystem shares some inputs and coordinates with the others.

Testing controllability in Eq. (4) of subsystems $\sum_i$ requires a computation of Lie brackets of generators $\boldsymbol{g}_i^j$ and their descendants for each subsystem separately. The check is slightly more complicated as functions describing generators depend on a position of the fixed joint. Nevertheless, it was checked[10] that condition in Eq. (4) is satisfied at any configuration and the model is strongly stratified controllable.

The second model described an underactuated (nonholonomic) satellite servicing robot equipped with two arms placed symmetrically atop of a relatively massive base as depicted in Fig. 1(b). For this

https://doi.org/10.1017/S0263574714002859 Published online by Cambridge University Press

robot, the following values of parameters $m_1 = m_2 = m_3 = m_4 = 1, m_0 = 15, l_1 = l_2 = l_3 = l_4 = 1$, $a = b = 1$ lead to the equations on the highest stratum $\mathcal{S}_0$

$$\begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{\theta} \end{pmatrix} = \dot{\boldsymbol{q}} = \sum_{i=1}^{4} \boldsymbol{g}_i(\boldsymbol{q}) u_i = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{u} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ A_1(\boldsymbol{q}) & A_2(\boldsymbol{q}) & A_3(\boldsymbol{q}) & A_4(\boldsymbol{q}) \end{bmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix}, \tag{10}$$

where

$$\begin{aligned} A_i(\boldsymbol{q}) &= f_i(\boldsymbol{q})/f_\theta(\boldsymbol{q}), \quad i = 1, \ldots, 4 \\ f_1(\boldsymbol{q}) &= -(171\, c_1 + 210\, c_2 + 57\, c_{12} + 27\, c_{13} + 9\, c_{123} + 9\, c_{134} + 3\, c_{1234} + 464) \\ f_2(\boldsymbol{q}) &= -(105\, c_2 + 57\, c_{12} + 9\, c_{123} + 3\, c_{1234} + 130) \\ f_3(\boldsymbol{q}) &= (171\, c_3 + 27\, c_{13} + 9\, c_{123} + 210\, c_4 + 57\, c_{34} + 9\, c_{134} + 3\, c_{1234} + 464) \\ f_4(\boldsymbol{q}) &= (105\, c_4 + 57\, c_{34} + 9\, c_{134} + 3\, c_{1234} + 130) \\ f_\theta(\boldsymbol{q}) &= (342\, c_1 + 210\, c_2 + 342\, c_3 + 210\, c_4 + 114\, c_{12} + 54\, c_{13} + 114\, c_{34} + \\ &\quad + 18\, c_{123} + 18\, c_{134} + 6\, c_{1234} + 1726). \end{aligned} \tag{11}$$

It is assumed that simultaneous actuation of all joints is forbidden and only two or three ones are actuated (unactuated joints remain fixed at their current positions). This assumption leads to a stratification with two levels of strata $\mathcal{S}_i$, $i = 1, 2, 3, 4$ with a fixed ith joint and $\mathcal{S}_{ij}$, $i, j = 1, \ldots, 4$, $i < j$ with fixed ith and jth joints. Subsystems defined on the first level are described by equations

$$\mathcal{S}_i : \qquad \dot{\boldsymbol{q}}|_{\mathbb{T}\mathcal{S}_i} = \boldsymbol{g}^i_j(\boldsymbol{q}|_{\mathcal{S}_i})u_j + \boldsymbol{g}^i_k(\boldsymbol{q}|_{\mathcal{S}_i})u_k + \boldsymbol{g}^i_l(\boldsymbol{q}|_{\mathcal{S}_i})u_l, \quad i = 1, \ldots, 4, \tag{12}$$

where i is the stratum index (and also the index of an unactuated joint) while j, k, and l denote indices of actuated joints. Similarly, subsystems defined on the second level of stratification satisfy the following equations:

$$\mathcal{S}_{ij} : \qquad \dot{\boldsymbol{q}}|_{\mathbb{T}\mathcal{S}_{ij}} = \boldsymbol{g}^{ij}_s(\boldsymbol{q}|_{\mathcal{S}_{ij}})u_s + \boldsymbol{g}^{ij}_k(\boldsymbol{q}|_{\mathcal{S}_{ij}})u_k, \quad i, j, s, k = 1, \ldots, 4, \ i < j, s < k, \{s, k\} \neq \{i, j\}. \tag{13}$$

It can be checked[10] that the presented stratified system is fully controllable at each stratum, thus the two-arm manipulator is strongly stratified controllable.

4. Nonholonomic Motion Planning

While solving a stratified motion planning task, it is necessary to solve a sequence of sub-tasks, each of them expressed as nonholonomic motion planning for a given subsystem. Many methods have been designed for the latter task[1,11] and are mainly for obstacle-free environments and for systems with extra features (flatness,[12] nilpotency,[13] chain form[14]). In obstacle cluttered environments additional problems in planning are encountered due to inadmissible areas in the configuration space. The constraints impact different methods of motion planning diversely. As a rule, local methods are not very sensitive to obstacles as they plan a motion in a small range, so obstacles only decrease the cone of admissible motion directions. Some global methods allow obstacles to be taken into account either as constraints or include them into an auxiliary quality function that penalizes any approaching to obstacles. Global methods also exist that work only in obstacle-free environments.

In this section, two general-purpose motion planning methods will be recalled. Global methods are represented by the Newton algorithm[15] while the Lie-algebraic method represents local ones.[11]

Both methods will be utilized later on as components of a stratified motion planner. Both methods are able to plan motion of a driftless nonholonomic system

$$\dot{\boldsymbol{q}} = \sum_{i=1}^{m} \boldsymbol{g}_i(\boldsymbol{q})u_i = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{u}, \qquad \boldsymbol{q} \in \mathbb{Q}, \quad \boldsymbol{u} \in \mathbb{L}^2_m[0, T], \tag{14}$$

where $\boldsymbol{q}$ is a configuration, $\boldsymbol{u}$ is the input of the system and $dim(\boldsymbol{q}) = n > m = dim(\boldsymbol{u})$.

4.1. Newton algorithm

The Newton algorithm based on an endogenous space approach[15] is founded on the concept of instantaneous kinematics

$$\boldsymbol{k}_{\boldsymbol{q}_0,T} : \mathbb{L}^2_m[0, T] \longrightarrow \mathbb{R}^n, \qquad \boldsymbol{k}_{\boldsymbol{q}_0,T}(\boldsymbol{u}(\cdot)) = \boldsymbol{\varphi}_{\boldsymbol{q}_0,T}(\boldsymbol{u}(\cdot)), \tag{15}$$

where $\boldsymbol{\varphi}_{\boldsymbol{q}_0,T}$ is the end-configuration of the flow of a system in Eq. (15) initialized at $\boldsymbol{q}_0$ when controls $\boldsymbol{u}(\cdot)$ are applied on the time interval $[0, T]$. Using kinematics in Eq. (15), a linear system that approximates behavior of the original one in Eq. (14), can be defined

$$\delta\dot{\boldsymbol{q}} = \frac{d}{dt}\mathrm{D}\boldsymbol{k}_{\boldsymbol{q}_0,t}(\boldsymbol{u}(\cdot))\delta\boldsymbol{u}(\cdot) = \boldsymbol{A}(t)\mathrm{D}\boldsymbol{k}_{\boldsymbol{q}_0,t}(\boldsymbol{u}(\cdot))\delta\boldsymbol{u}(t) + \boldsymbol{B}(t)\delta\boldsymbol{u}(t) = \boldsymbol{A}(t)\delta\boldsymbol{q} + \boldsymbol{B}(t)\delta\boldsymbol{u}, \tag{16}$$

where

$$\boldsymbol{A}(t) = \frac{\partial(\boldsymbol{G}(\boldsymbol{q}(t))\boldsymbol{u}(t))}{\partial\boldsymbol{q}}, \quad \boldsymbol{B}(t) = \boldsymbol{G}(\boldsymbol{q}(t)),$$

and $\delta\boldsymbol{u}(t)$ denotes a small variation of controls $\boldsymbol{u}(\cdot)$ at time t. According to Eq. (16), a nonholonomic Jacobian matrix $\mathrm{D}\boldsymbol{k}_{\boldsymbol{q}_0,t}$ transforms the variation of controls $\delta\boldsymbol{u}$ into a variation of configuration $\delta\boldsymbol{q}$. Solution of Eq. (16) is in the form

$$\delta\boldsymbol{q}(T) = \int_{s=0}^{T} \boldsymbol{\Phi}(T, s)\boldsymbol{B}(s)\delta\boldsymbol{u}(s)\, ds, \tag{17}$$

where $\boldsymbol{\Phi}(T, s)$ is a fundamental matrix satisfying $d(\boldsymbol{\Phi}(T, s))/dt = \boldsymbol{A}(t)\boldsymbol{\Phi}(T, s)$ with an initial condition $\boldsymbol{\Phi}(s, s) = Identity$. Usually, controls are represented in the form of a finite series and an orthogonal basis $\boldsymbol{\phi}(\cdot)$ is selected to describe controls as a linear combination of some of its items

$$u_k(t) = \sum_{j=1}^{N^k} \lambda^k_j \phi^k_j(t), \quad k = 1, \ldots, m, \quad t \in [0, T], \tag{18}$$

where N^k is the number of items selected to express the kth control. Vector

$$\boldsymbol{\lambda}^k = (\lambda^k_1, \lambda^k_2, \ldots, \lambda^k_{N_k})^T, \qquad k = 1, \ldots, m,$$

collects coefficients of the kth control in Eq. (18) and

$$\boldsymbol{\Lambda} \ni \boldsymbol{\lambda} = (\boldsymbol{\lambda}^1, \ldots, \boldsymbol{\lambda}^m)^T,$$

gathers coefficients for all controls. The search of controls is performed within the space of their coefficients $\boldsymbol{\Lambda} \ni \boldsymbol{\lambda}$ as the time-dependent basis functions $\boldsymbol{\phi}(\cdot)$ are fixed.

After substituting Eqs. (18) into (17), we get

$$\delta\boldsymbol{q}(T) = \boldsymbol{J}_{q_0,T}(\boldsymbol{\lambda}_i)\delta\boldsymbol{\lambda}, \tag{19}$$

https://doi.org/10.1017/S0263574714002859 Published online by Cambridge University Press

and vector $\boldsymbol{\lambda}$ is changed according to an iterative formula introduced by the basic Newton algorithm for nonholonomic systems[15]

$$\boldsymbol{\lambda}_{i+1} = \boldsymbol{\lambda}_i + \gamma \cdot \boldsymbol{J}^{\#}_{\boldsymbol{q}_0,T}(\boldsymbol{\lambda}_i)(\boldsymbol{q}_d - \boldsymbol{k}_{\boldsymbol{q}_0,T}(\boldsymbol{u}(\cdot,\boldsymbol{\lambda}_i))), \tag{20}$$

where γ is a given positive constant, $\boldsymbol{k}_{\boldsymbol{q}_0,T}$ represents system kinematics, defined by Eq. (15), $\boldsymbol{q}_d$ is a goal configuration and $\boldsymbol{J}^{\#}_{\boldsymbol{q}_0,T} = \boldsymbol{J}^T_{\boldsymbol{q}_0,T}(\boldsymbol{J}_{\boldsymbol{q}_0,T}\boldsymbol{J}^T_{\boldsymbol{q}_0,T})^{-1}$ is the pseudo matrix inversion.[16]

4.2. Lie-algebraic method

In Lie-algebraic terms,[11] a local, around a given configuration $\boldsymbol{q}$, evolution $\delta\boldsymbol{q}$ of the system state under action of controls $\boldsymbol{u}(\cdot)$ can be expressed as follows:

$$\delta\boldsymbol{q} = z(t)(\boldsymbol{q}) \sim \alpha_1(\boldsymbol{u}(\cdot))B_1(\boldsymbol{q}) + \alpha_2(\boldsymbol{u}(\cdot))B_2(\boldsymbol{q}) + \alpha_3(\boldsymbol{u}(t))B_3(\boldsymbol{q}) + \cdots \tag{21}$$

where α's are integral, control dependent coefficients and B's are elements of the Ph. Hall basis[11] (i.e. an independent set of vector fields derived from generators $\boldsymbol{g}_i$ and their descendants using the Lie bracket operation)is evaluated at $\boldsymbol{q}$. Locality means that the formula of Eq. (21) is valid only locally, around current configuration $\boldsymbol{q}$, and the trajectory initialized at $\boldsymbol{q}$ does not move too far away from the configuration. Consequently, only a very few first elements of the infinite sequence described by rhs of Eq. (21) can be taken and neglected terms cannot significantly disturb the desired motion. At least $r \geq n$ items have to be taken to preserve (small time local) controllability[5] of the system around $\boldsymbol{q}$. The desired motion to the goal $\boldsymbol{q}_d$ can be defined as $\delta\boldsymbol{q} = \gamma(\boldsymbol{q}_d - \boldsymbol{q})$ with a small positive constant γ. An inverse kinematic problem to find controls $\boldsymbol{u}(\cdot)$ resulting in a given state transfer $\delta\boldsymbol{q}$ is defined by Eq. (21). Similar to the Newton algorithm considered previously, a parametrization of controls, given by Eq. (18), is applied.

Now, with fixed basis functions $\boldsymbol{\phi}(\cdot)$, the local system state evolution in Eq. (21) can be expressed as

$$\gamma(\boldsymbol{q}_d - \boldsymbol{q}) = \delta\boldsymbol{q} = \sum_{i=1}^{r} \alpha_i(\boldsymbol{\lambda})B_i(\boldsymbol{q}), \tag{22}$$

with known kinematics

$$(\alpha_1, \ldots, \alpha_r)^T = \boldsymbol{\alpha} = \boldsymbol{h}(\boldsymbol{\lambda}), \tag{23}$$

and coefficients of controls $\boldsymbol{\lambda}$ to be determined. From Eq. (22), using matrix pseudo-inverse, the desired values of $\boldsymbol{\alpha_d} = (\alpha_1, \ldots, \alpha_r)^T$ are derived. Inverse kinematics for kinematics in Eq. (23) is solved, once again, with the Newton algorithm

$$\boldsymbol{\lambda}_{i+1} = \boldsymbol{\lambda}_i + \gamma \cdot \boldsymbol{J}^{\#}(\boldsymbol{\lambda}_i)(\boldsymbol{\alpha}_d - \boldsymbol{h}(\boldsymbol{\lambda}_i)), \tag{24}$$

where $\boldsymbol{J} = \partial\boldsymbol{h}/\partial\boldsymbol{q}$ and the initial value of $\boldsymbol{\lambda}_0$ is assumed.

4.3. Concatenation of controls and hybridization of algorithms

Stratified motion planning will require the solving of a sequence of sub-tasks, sometimes for different models and using local or global methods. Moreover, some controls can be shared between consecutive planning. Therefore, problems of hybridization of methods and concatenation of controls arise. Hybridization of methods allows the advantages of local and global methods to be exploited. Both of the tasks can be solved using redundancy in representation of controls in Eq. (18) and kinematics in Eqs. (15) and (23), to determine a useful system behavior as a function of coefficients of controls $\boldsymbol{\lambda}$. Note that the representation of controls can vary from one sub-planning to the other and redundancy in controls is also used for a single planning to simplify the preserving of controllability of the system in Eq. (14). On shared and unshared controls between consecutive sub-plannings additional constraints can be added to get their desired properties (C^k continuity of controls at boundaries or their vanishing at boundaries). Each constraint on controls takes away one degree of freedom. Therefore, a representation of controls in Eq. (18) has to be rich enough to solve a motion planning task.

Fortunately, controls depend linearly on the parameters $\boldsymbol{\lambda}$. Consequently, each constraint on a value or a derivative of controls at a particular point of time shares this characteristic and all constraints can be coupled into a matrix equation

$$\boldsymbol{C} \cdot \boldsymbol{\lambda} = \boldsymbol{c}, \quad \dim \boldsymbol{\lambda} = r. \tag{25}$$

If r_1 denotes the number of independent constraints, the constant $(r_1 \times r)$ non-singular matrix $\boldsymbol{C}$ depends on the representation of controls while $(r_1 \times 1)$ vector $\boldsymbol{c}$ fixes the boundary values of controls.

The parametrization of controls should be selected carefully. An appropriate number and quality of basic functions for each control have to be chosen to satisfy the requirements imposed. It is easy to deduce that $\mathbb{C}^2$ continuity cannot be satisfied with controls in Eq. (18) composed of even functions only, as discontinuity of the first derivative of controls appears in a typical case.

As the matrix $\boldsymbol{C}$ is non-singular, the vector $\boldsymbol{\lambda}$, can be decomposed into two sets of dependent/independent variables $\boldsymbol{\lambda}_{dep}, \boldsymbol{\lambda}_{ind}$ ($\dim \boldsymbol{\lambda}_{dep} = r_1, \dim \boldsymbol{\lambda}_{ind} = r_2 = r - r_1$). Using the Gauss elimination procedure applied to Eq. (25), all variables $\boldsymbol{\lambda}_{dep}$ can be expressed as a function of variables $\boldsymbol{\lambda}_{ind}$

$$\boldsymbol{\lambda}_{dep} = \tilde{\boldsymbol{C}} \cdot \boldsymbol{\lambda}_{ind} + \tilde{\boldsymbol{c}}. \tag{26}$$

It is worth noting that there exists some flexibility in selection of independent variables. Taking the time derivative of Eq. (26), an equality in small variations is obtained

$$\delta \boldsymbol{\lambda}_{dep} = \tilde{\boldsymbol{C}} \cdot \delta \boldsymbol{\lambda}_{ind}. \tag{27}$$

Constraints in Eq. (27) can be easily injected into the Newton algorithm. Below, derivations will be presented for the Newton algorithm based on the endogenous space approach (similarly, it also works for the Lie-algebraic method). After substituting Eq. (19) into a general parametric Newton scheme, the algorithm takes the form

$$\delta \boldsymbol{q}(T) = \boldsymbol{J}_{\boldsymbol{q}_0,\boldsymbol{T}}(\boldsymbol{u}(\cdot, \boldsymbol{\lambda})) \cdot \boldsymbol{R} \cdot \begin{bmatrix} \boldsymbol{I}_{r_2} \\ \tilde{\boldsymbol{C}} \end{bmatrix} \cdot \delta \boldsymbol{\lambda}_{ind} = \tilde{\boldsymbol{J}}_{\boldsymbol{q}_0,\boldsymbol{T}}(\boldsymbol{u}(\cdot, \boldsymbol{\lambda}_{ind})) \cdot \delta \boldsymbol{\lambda}_{ind}, \tag{28}$$

where $\boldsymbol{I}_{r_2}$ is $(r_2 \times r_2)$ the identity matrix and $(r \times r)$ matrix $\boldsymbol{R}$ (mostly composed of zeroes) rearranges coordinates to enumerate independent variables from the set $\boldsymbol{\lambda}_{ind}$ in a natural order $(1, 2 \ldots, r_2)$

$$\boldsymbol{\lambda} = \boldsymbol{R} \cdot \begin{pmatrix} \boldsymbol{\lambda}_{ind} \\ \boldsymbol{\lambda}_{dep} \end{pmatrix}. \tag{29}$$

With new independent variables $\boldsymbol{\lambda}_{ind}$, the algorithm in Eqs. (20) or (24) can be run to solve the current planning task while boundary values result either from a previous planning or from demanded values at the end of the current planning.

It should be pointed out that the number of components in $\boldsymbol{\lambda}_{ind}$ has to be at least equal to the dimensionality of the state space $\mathbb{Q}$. However, a small redundancy in independent variables is advised.

5. Stratified Motion Planning

Motion planning of stratified nonholonomic systems is an extension of the task for continuous nonholonomic systems.[11] Therefore, the task formulation and main ingredients are very similar for both tasks. The main difference results from a stratification of the configuration manifold, thus the initial and the final configurations are defined rather on a union of all strata available than within a configuration space (considered as a single admissible stratum). Moreover, while planning a motion, switching of strata is likely to occur. The continuous nonholonomic motion planning task is difficult by itself, and stratification complicates the problem even further. In order to point out the main

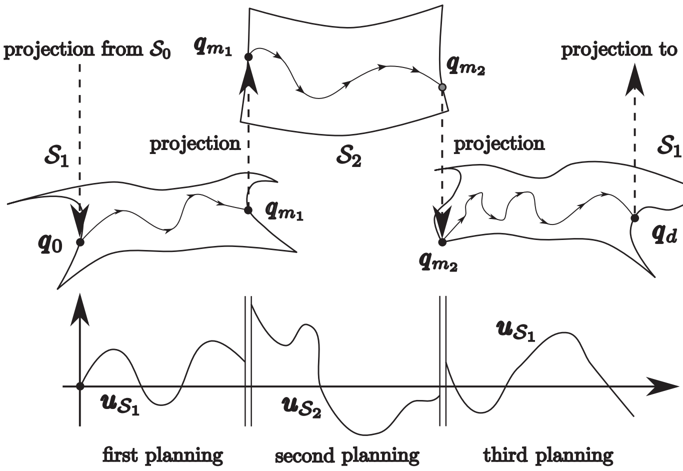

Fig. 2. Stratified motion planning.

difficulties related to the stratified task no additional constraints will be added and the stratified task is stated in a collision-free configuration space without restrictions on controls.

In general, stratified motion planning requires planning of not only a motion of underactuated subsystems but also dispatchment of sub-tasks for subsystems as the global task is defined within the highest, unavailable stratum ($\mathcal{S}_0$) with dimensionality larger than for any of the subsystems. Consequently, switching between subsystems is indispensable. The prescribed global task is to be decomposed into sub-tasks defined for subsystems, and then solved subsequently. The particular sequence of selected strata ($\mathcal{S}_I, \ldots, \mathcal{S}_{mid}, \ldots, \mathcal{S}_D$) uniquely determines the subsystems (possibly, each of them can be selected many times) and naturally defines a list of sub-tasks to solve.

The idea of stratified motion planning is illustrated in Fig. 2. At first, the sequence of strata is chosen as $\mathcal{S}_1 \to \mathcal{S}_2 \to \mathcal{S}_1$. Then, the initial given configuration is projected from $\mathcal{S}_0$ onto $\mathcal{S}_1$. Moving within stratum $\mathcal{S}_1$, the first planning sets the desired values for only a few coordinates (because $\forall_i n_i < n_0$) and completes at configuration $\boldsymbol{q}_{m_1}$. After switching into stratum $\mathcal{S}_2$, a different set of coordinates can obtain their desired values. Then, if needed, a stratum can be changed again to steer coordinates that are not defined on stratum $\mathcal{S}_2$.

Each planning can be viewed as a separate task governed by its own model with initial and final configurations appropriately defined for the current stratum. Although possible, separation of sub-tasks should not be considered as the best solution. It simplifies the global task, but ignores the internal relationships between the subsystems. This omission may result in controls with bad properties (for example, non-continuous). Consequently, a proper concatenation of controls should distinguish between exclusive and shared controls among neighboring strata, and properly set their values either to zero or to required ones, respectively, while switching.

5.1. Stratified motion planner

The algorithm of stratified motion planning should take initial and final configurations defined on the highest stratum, $\boldsymbol{q}_0, \boldsymbol{q}_d \in \mathcal{S}_0$, and compute controls steering the system between those configurations. Moreover, some regularity conditions on resulting controls can be imposed.

The diagram of the proposed algorithm is presented in Fig. 3. Controls are chosen from available inputs for subsystems, therefore the planning is performed as a sequence of sub-tasks stated for subsystems. A proper sequencing of sub-tasks (thus strata and subsystems) is solved by *a task decomposer module* described in details later on. The sequence of tasks is processed by *a sequential motion planner module*: each task is taken from the *task queue* and solved by *a nonholonomic motion planner* using one from the available motion planning algorithms, possibly, preserving C^k continuity of controls (cf. Fig. 3). To satisfy the requirement, each sub-planning should take into account the results of the previous sub-planning and also data for the next planning. Because a nonholonomic motion planner performs planning on a single stratum, it requires not only the initial and final configurations but also a model valid for the current planning. This model is uniquely defined by those coordinates of global configuration $\boldsymbol{q}$ which are fixed during the current planning. When all

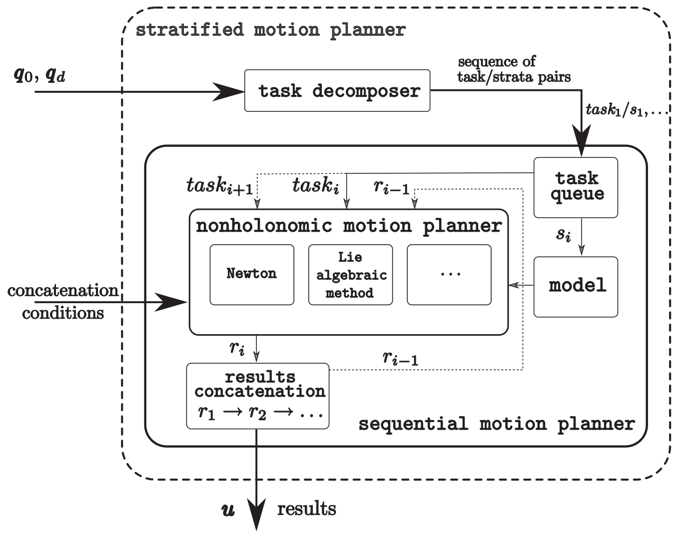

Fig. 3. Stratified motion planner.

the tasks from the task queue are solved, their results are collected and the global motion planning is completed. Below the modules will be described with details.

5.2. *Nonholonomic motion planner*

Although a nonholonomic motion planner module uses algorithms implemented for continuous nonholonomic systems, the C^k concatenation of controls sets additional constraints. Available controls for the current ith planning can be divided into the following two groups:

1. Those shared with $(i-1)$st planning should get values (and derivatives) at the initial time of the ith planning determined by controls from the previous planning. The remaining controls should be initiated with zero values.
2. Those unshared with $(i+1)$st planning should get zero values (and derivatives) at the final time of the ith planning. The remaining controls are unrestricted.

A nonholonomic motion planner module implements the methods described in Section 4, allowing for the concatenation and hybridization of methods. Practically valuable concatenation conditions are collected below:

- Each planning is independent; no conditions are imposed on joining pieces of controls.
- Left-continuous, l-$\mathbb{C}^0$ controls are desirable, $\boldsymbol{u}^i(0) = \boldsymbol{u}^{i-1}(T_{i-1})$.
- Left-differentiable, l-$\mathbb{C}^1$ controls are guaranteed, $\boldsymbol{u}^i(0) = \boldsymbol{u}^{i-1}(T_{i-1})$ and $\dot{\boldsymbol{u}}^i(0) = \dot{\boldsymbol{u}}^{i-1}(T_{i-1})$.
- Left-right-continuous, lr-$\mathbb{C}^0$ controls are desirable, $\boldsymbol{u}^i(0) = \boldsymbol{u}^{i-1}(T_{i-1})$ and $\boldsymbol{u}^i(T_i) = \boldsymbol{u}^{i+1}(0)$.
- Left-right-differentiable, lr-$\mathbb{C}^1$ controls are guaranteed, $\boldsymbol{u}^i(0) = \boldsymbol{u}^{i-1}(T_{i-1})$, $\dot{\boldsymbol{u}}^i(0) = \dot{\boldsymbol{u}}^{i-1}(T_{i-1})$ and $\boldsymbol{u}^i(T) = \boldsymbol{u}^{i+1}(0)$, $\dot{\boldsymbol{u}}^i(T) = \dot{\boldsymbol{u}}^{i+1}(0)$.
- 'Soft start' and 'full stop' features, $\boldsymbol{u}(0) = 0$, $\boldsymbol{u}(T) = 0$, $\dot{\boldsymbol{u}}(0) = 0$, $\dot{\boldsymbol{u}}(T) = 0$.

5.3. *Task decomposer*

The purpose of the module is to translate the initial task defined on $\mathcal{S}_0$, for (inadmissible) system Σ_0, into a sequence of stratified sub-tasks. The algorithm for the module runs as follows:

Step 1. Compare initial configuration $\boldsymbol{q}_0$ with final configuration $\boldsymbol{q}_d$ to determine coordinates that differ and name the essential coordinated.

Step 2. Determine the strata on which the motion planning task is defined. The only condition the strata should satisfy is that the union of their coordinates includes all essential coordinates. Usually, the selection of strata is not unique and it can be optimized.

Step 3. Form some sequences from the selected strata. In each sequence any stratum can be used more than once.

Step 4. For each selected sequence determine the boundary configurations for the sub-tasks and solve the motion planning sub-tasks sequentially.

Step 5. As a final solution output the sequence (trajectory, controls) which is the best among sequences is considered.

In Step 4, a conservative approach prompts the desired values of essential coordinates to be obtained as soon as possible. For example, if at stratum $\mathcal{S}_a$ one can act on essential coordinates q_1, q_2, their goal values for the planning on $\mathcal{S}_a$ are set to the required from configuration $\boldsymbol{q}_d$ and other ones should remain unchanged. When a postponed approach is preferred and if there exists in the selected sequence of strata a stratum $\mathcal{S}_b$ (placed somewhere after stratum $\mathcal{S}_a$ in the sequence) which acts on the essential coordinate q_2 , there is no necessity to set its goal value while planning on $\mathcal{S}_a$ and do the task while planning on stratum $\mathcal{S}_b$.

It is difficult to judge which strategy is the (sub-) optimal one. In practical situations there is not too much strata and their sequences. Therefore, many (all) acceptable sequences of strata can be checked and evaluated based on some criteria (energy expenditure, amplitude of maneuvers, etc.)

As an example, let us consider a one-arm space manipulator moving between $\boldsymbol{q}_0 = (0, \pi/4, 0, 0)$ and $\boldsymbol{q}_d = (-\pi/3, \pi/3, \pi/4, -\pi/12)$. Values of identical coordinates in $\boldsymbol{q}_0$ and $\boldsymbol{q}_d$ differ, so all coordinates are essential. From the system stratification it is known that q_1 can be changed via $\mathcal{S}_2$ or $\mathcal{S}_3$. Similarly, changing q_2 requires motion either within $\mathcal{S}_1$ or $\mathcal{S}_3$ and q_3 can be impacted via $\mathcal{S}_1$ or $\mathcal{S}_2$. Steering q_4 is possible on any stratum. On the other hand each stratum blocks only one coordinate, so within $\mathcal{S}_1$ only the coordinates q_2, q_3, q_4 can be changed.

To steer all essential coordinates, a single stratum is not enough. A sequence of planning has to be composed of at least two different strata. For instance $(\mathcal{S}_1, \mathcal{S}_2)$, $(\mathcal{S}_2, \mathcal{S}_1)$, $(\mathcal{S}_2, \mathcal{S}_3)$, $(\mathcal{S}_3, \mathcal{S}_2)$, $\ldots$, $(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_2)$, $\ldots$, $(\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_1)$, $\ldots$. The task decomposer module should prefer short sequences, although a redundant sequence could also improve the character of motion, as shown in Task 4 in the simulation section.

For a given sequence of strata, a choice of boundary configurations for each subtask is performed sequentially: each sub-task tries to set as many essential coordinates to their desired values as possible leaving the already steered one unchanged. For example, for the sequence $(\mathcal{S}_1, \mathcal{S}_2)$ boundary configurations follow: $\boldsymbol{q}_0 = (0, \pi/4, 0, 0) \rightarrow (0, \pi/3, \pi/4, -\pi/12) \rightarrow (-\pi/3, \pi/3, \pi/4, -\pi/12) = \boldsymbol{q}_d$.

6. Simulations

The stratified nonholonomic motion planner was tested on numerous tasks. In all simulations, the Newton algorithm and the Lie-algebraic method used the same subset of the Fourier basis to represent controls

$$u_k(t) = \lambda_0^k + \sum_{i=1}^{3} \left(\lambda_i^k \cdot \sin(2i\pi/T \cdot t) + \lambda_{i+3}^k \cos(2i\pi/T \cdot t) \right), \quad k = 1, \ldots, m, \tag{30}$$

where the superscript k assigns $\boldsymbol{\lambda}$'s to the kth control. The free-to-choose time horizon T of a single planning is advised not to be a multiple of 2π (otherwise sine functions vanish at boundaries and the number of variables in $\boldsymbol{\lambda}$ to satisfy additional requirements at boundaries of a single planning decreases almost twice. Later on the value of T was set to 2.17. When the Newton algorithm is invoked, each planning is performed on the time interval [0, 1] (finally, shifted accordingly if it is not the first planning in a sequence). The length of time interval for the Newton algorithm can be modified after solving a motion planning task using a time scaling technique. Post-planning time scaling is especially useful to decrease/increase amplitudes of controls at the expense of increasing/decreasing time T. For the Lie-algebraic method, the total interval of the whole sub-planning depends on the number of required steps performed, each step lasts 0.51. Arguments for setting the time interval T of a single planning for the Newton algorithm remains valid for the Lie-algebraic method also. For both algorithms in Eqs. (20) and (24), the accepted accuracy to reach a sub-goal was set to $0.001[rd]$. Energy of controls, $\int_0^T \sum_{i=1}^m u_i^2(t)\, dt$, and plots of trajectories and controls are provided.

At first, basic aspects of the stratified motion were illustrated. In each simulation the switching points and sequence of strata are given. It means that a benchmark scenario is realized. In this

https://doi.org/10.1017/S0263574714002859 Published online by Cambridge University Press

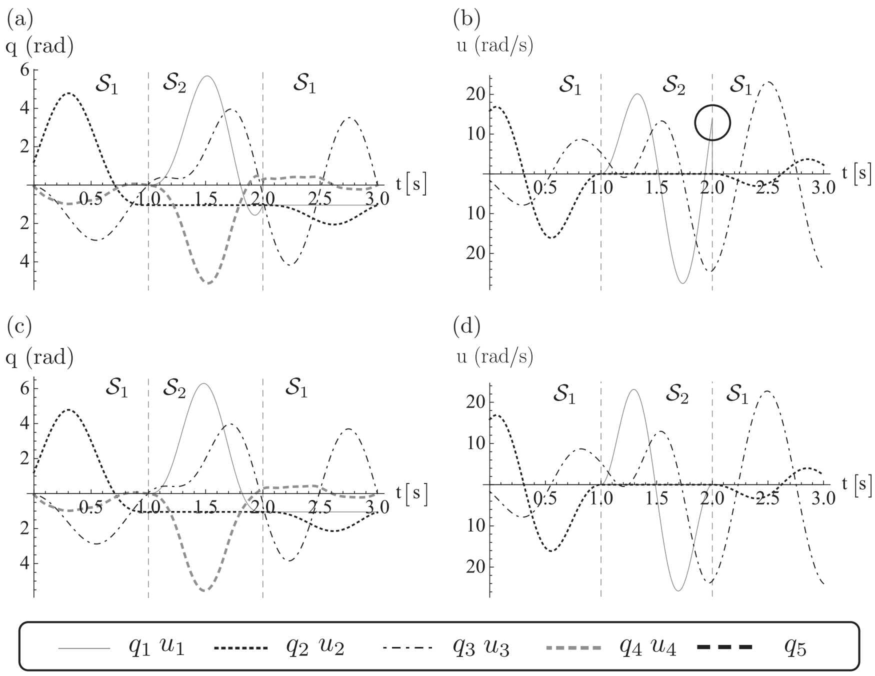

Fig. 4. Trajectory and controls for (a), (b) l-$\mathbb{C}^1$ concatenation; (c), (d) lr-$\mathbb{C}^1$ concatenation.

part, the concatenation method is being tested and different concatenation criteria are compared. Moreover, some scenarios test certain switching orders. All simulations in this group utilize the Newton algorithm for single motion planning.

The second group of tests concentrates on the hybridization of planning methods where the Newton algorithm and the Lie-algebraic method (representatives of global and local methods, respectively) are coupled.

Finally, the stratified motion planner was tested for a variety of tasks. The switching sequences of different lengths, and composed of the same strata but in a different order, were compared. All simulations are conducted using models of a one/two arm space manipulator.

The structure of each simulation follows: a task statement (initial and desired configurations, optionally, a given sequence of strata), a model of the stratified system, a description of the tested property, and also different variants of the task. The task statement is followed by a discussion of results, which includes some numeric results and plots of trajectory and controls.

Simulation 1: a long sequence of switching is tested on a one-arm space manipulator moving between the strata and configurations given

$$\begin{array}{ccccccc} \boldsymbol{q}_0 & \xrightarrow{\text{on } \mathcal{S}_1} & \boldsymbol{q}_{mid_1} & \xrightarrow{\text{on } \mathcal{S}_2} & \boldsymbol{q}_{mid_2} & \xrightarrow{\text{on } \mathcal{S}_1} & \boldsymbol{q}_d, \\ (0, \pi/3, 0, 0) & \to & (0, -\pi/3, 0, 0) & \to & (-\pi/3, -\pi/3, -\pi/6, \pi/12) & \to & (0, -\pi/3, 0, 0). \end{array}$$

using l-$\mathbb{C}^1$ and lr-$\mathbb{C}^1$ controls. Energy on controls for the two variants were equal to 817.50 and 812.04, respectively. Plots of trajectory and controls are presented in Fig. 4. Although energy spent on controls is almost the same, a difference can be spotted in Figs. 4(b) and (d). In the first case, the algorithm ignores the fact that u_1 will not be used in the third planning (on stratum $\mathcal{S}_1$) and allows it to change freely and consequently control u_1 is discontinuous. When lr-$\mathbb{C}^1$ controls were applied the continuity of controls was retrieved.

https://doi.org/10.1017/S0263574714002859 Published online by Cambridge University Press

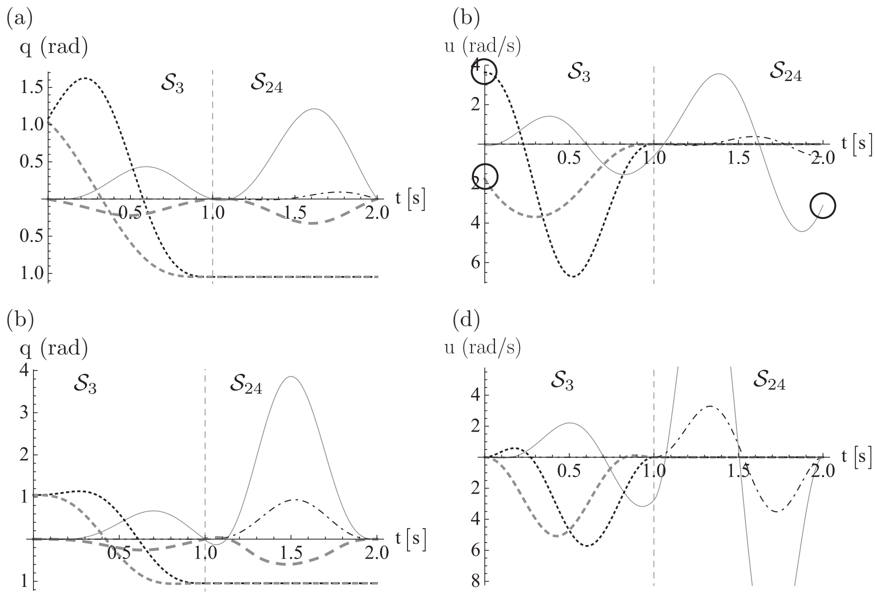

Fig. 5. Trajectory and controls for planning (a), (b) with and (c), (d) without soft start and full stop properties.

Simulation 2: on a two-arm manipulator, a multi-level stratified model was tested with and without soft-start and full-stop properties. The task was to perform the following scenario:

$$\begin{array}{ccccc} \boldsymbol{q}_0 & \xrightarrow{\text{on } \mathcal{S}_3} & \boldsymbol{q}_{mid} & \xrightarrow{\text{on } \mathcal{S}_{24}} & \boldsymbol{q}_d, \\ (0, \pi/3, 0, \pi/3, 0) & \rightarrow & (0, -\pi/3, 0, -\pi/3, 0) & \rightarrow & (0, -\pi/3, 0, -\pi/3, 0). \end{array}$$

The resulting trajectory and controls are presented in Fig. 5 (some parts of the controls in Fig. 5(d) were truncated to preserve the readability of the rest.

With no soft start, the energy of controls was equal to 30.41 and increased to 113.34 when a soft start was required. It can be observed that additional constraints are likely to increase the total energy of controls. Nonetheless, from a practical point of view, the controls presented in Fig. 5(b) are of limited use as the large-value controls should be suddenly switched on or off (see encircled points at the beginning and the end of planning) disturbing real motion. This simulation also illustrates a common feature of stratified concatenations: while switching strata the dimensionality of subsystems and the number of their inputs may change. In this particular case, on stratum $\mathcal{S}_3$, $\dim(\mathcal{S}_3) = 4$, $m_3 = 3$ while on stratum $\mathcal{S}_{24}$, $\dim(\mathcal{S}_{24}) = 3$, $m_{24} = 2$. The planner dealt with this situation properly and all controls had the desired properties.

Simulation 3: illustrates a hybridization which allows the mixing of different planning methods in a single sequence. For the scenario of motion

$$\begin{array}{ccccccc} \boldsymbol{q}_0 & \xrightarrow{\text{Newton on } \mathcal{S}_1} & \boldsymbol{q}_{mid_1} & \xrightarrow{\text{Newton on } \mathcal{S}_2} & \boldsymbol{q}_{mid_2} & \xrightarrow{\text{Lie on } \mathcal{S}_1} & \boldsymbol{q}_d, \\ (0, \pi/3, 0, 0) & \rightarrow & (0, -\pi/3, 0, 0) & \rightarrow & (-\pi/3, -\pi/3, \pi/6, \pi/12) & \rightarrow & (0, 0.2, 0.1, -0.1) + \boldsymbol{q}_{mid_2}, \end{array}$$

the Newton algorithm was used to plan a motion across two strata and (when the desired configuration is reached) a slight configuration adjustment was made with the local Lie-algebraic method. The task was solved using lr-$\mathbb{C}^1$ controls. The energy expenditure on the controls was 244.00. Figure 6 presents the resulting trajectory and controls while in Fig. 7 the stroboscopic view of global and local motions at consecutive stages is visualized. As shown in Fig. 6, the hybridization works properly and the concatenation criterion is met. Moreover, the periodic local motion limits the amplitude of configuration change. The algorithm correctly recognizes the used controls and restraints of the non-actuated joints. The important difference between the Newton and Lie-algebraic algorithms can be seen in details in Fig. 7. Being local, the Lie-algebraic method plans a motion with only small

https://doi.org/10.1017/S0263574714002859 Published online by Cambridge University Press

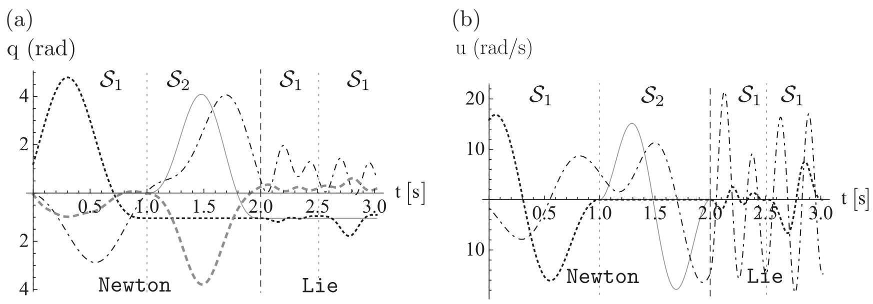

Fig. 6. Trajectory and controls.

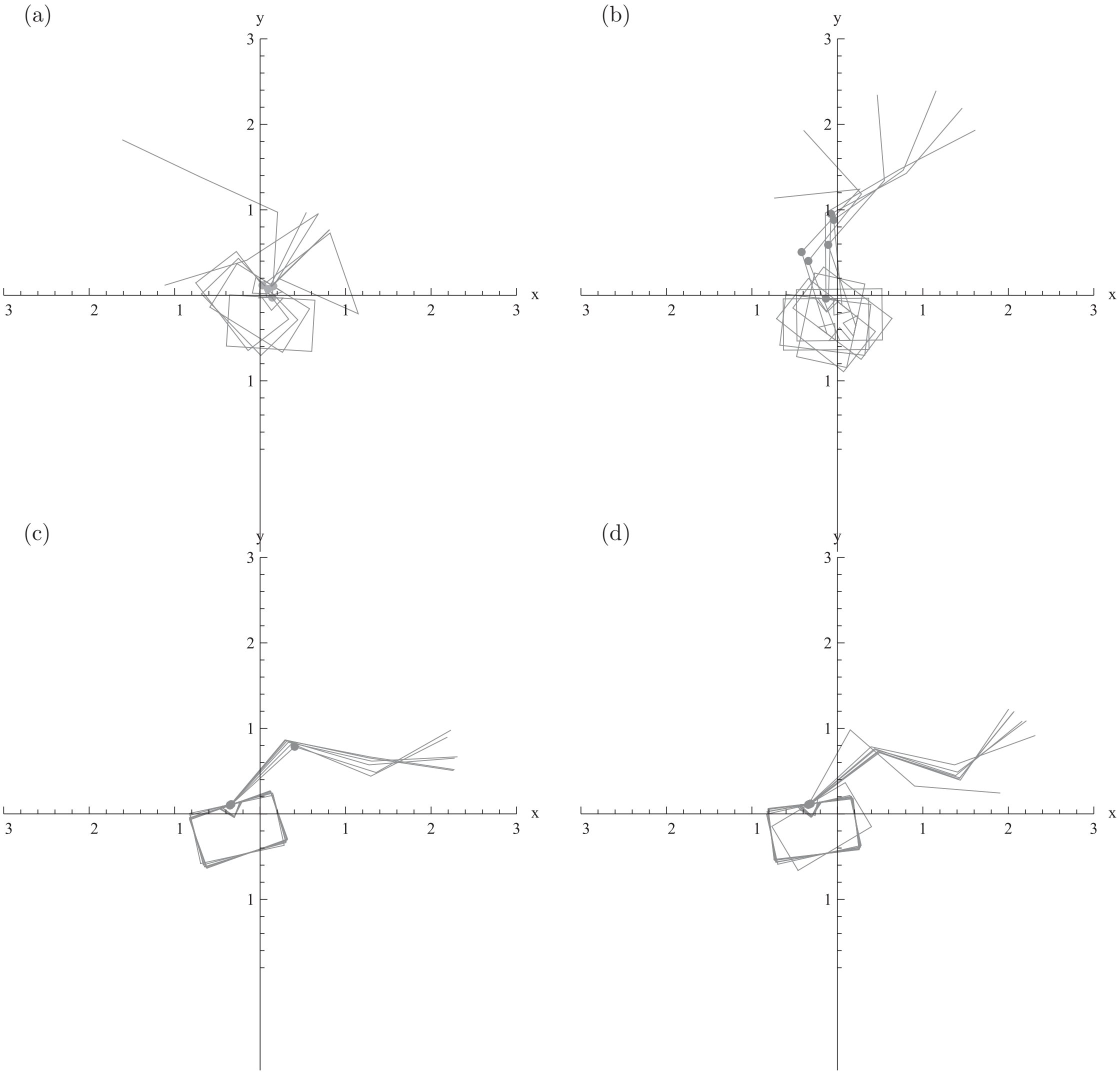

Fig. 7. Stroboscopic view of a robot's motion: (a), (b) two stages performed with the global Newton algorithm on stratum S_1 and S_2; (c), (d) two steps carried on stratum S_1 using the local, Lie-algebraic method.

replacements of the manipulator, cf. Figs. 7(c) and (d). On the other hand, the Newton method generates extensive motion of links, cf. Figs. 7(a) and (b), sometimes even when the initial and goal configuration of the sub-planning are not too far from each other.

Simulation 4: finally, after testing each feature of the stratified motion planner separately, its fully automatic work is illustrated. Tests were conducted for a one-arm space manipulator with boundary

https://doi.org/10.1017/S0263574714002859 Published online by Cambridge University Press

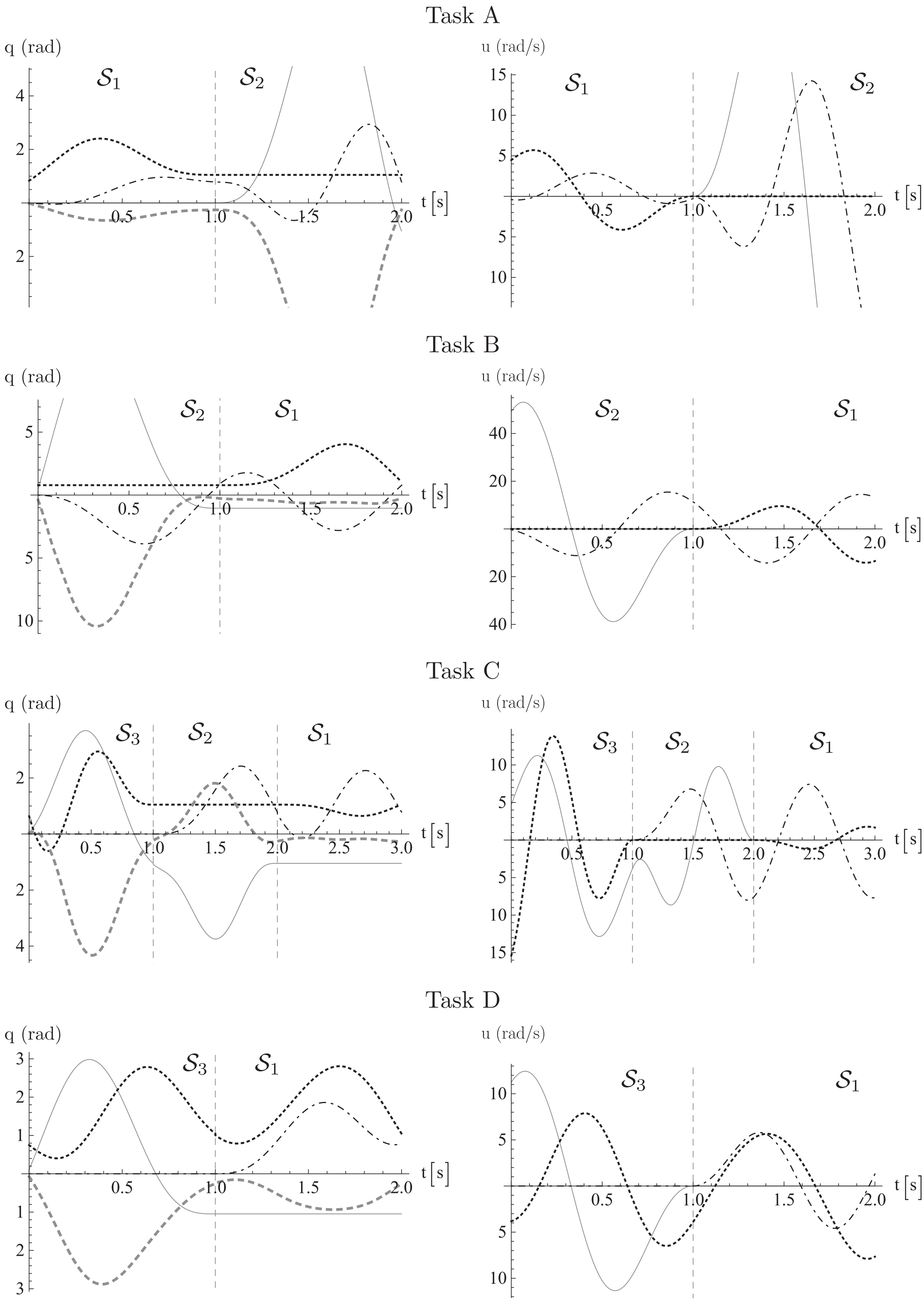

Fig. 8. Trajectories (left column) and controls (right column) for Tasks A–D.

https://doi.org/10.1017/S0263574714002859 Published online by Cambridge University Press

configurations $\boldsymbol{q}_0 = (0, \pi/4, 0, 0)$, $\boldsymbol{q}_d = (-\pi/3, \pi/3, \pi/4, -\pi/12)$ and lr-$\mathbb{C}^1$ controls demanded. The planning task was decomposed into sequences of strata and boundary configurations for consecutive sub-tasks were automatically generated with the stratified motion planner. The planner generated a few variants of possible solutions and some of them are presented below:

Task A: sequence $(\mathcal{S}_1, \mathcal{S}_2)$ with boundary configurations $\boldsymbol{q}_0 \to \boldsymbol{q}_{mid} \to \boldsymbol{q}_d$ with via-configuration $\boldsymbol{q}_{mid} = (0, \pi/3, \pi/4, -\pi/12)$.
Task B: sequence $(\mathcal{S}_2, \mathcal{S}_1)$, this time with mid-configuration $\boldsymbol{q}_{mid} = (-\pi/3, \pi/4, \pi/4, -\pi/12)$.
Task C: three-step sequence $(\mathcal{S}_3, \mathcal{S}_2, \mathcal{S}_1)$ with boundary configuration $\boldsymbol{q}_0 \to \boldsymbol{q}_{mid_1} \to \boldsymbol{q}_{mid_2} \to \boldsymbol{q}_d$, where $\boldsymbol{q}_{mid_1} = (-\pi/3, \pi/3, 0, -\pi/12)$, and $\boldsymbol{q}_{mid_2} = (-\pi/3, \pi/3, \pi/4, -\pi/12)$.
Task D: an alternative two-item sequence $(\mathcal{S}_3, \mathcal{S}_1)$ with mid-configuration $\boldsymbol{q}_{mid} = (-\pi/3, \pi/3, 0, -\pi/12)$.

The energy spent on controls for the four scenarios was equal to 542.36, 1212.92, 237.04, 124.59, respectively. The plots of resulting trajectories and controls are presented in Fig. 8.

Comparing the results of Tasks A and B, the importance of switching order is highlighted. Both sequences solved the planning task, however, the energy of controls for sequence $(\mathcal{S}_2, \mathcal{S}_1)$, was more than two times larger than for sequence $(\mathcal{S}_2, \mathcal{S}_1)$. It is an expected result as previous plannings set initial conditions for the current planning (boundary configurations as well as a model description for a current planning strongly depend on fixed configurations actively changed by previous plannings). A minimal number of strata in a sequence does not guarantee an energy efficient solution. The third sequence, is energetically more efficient than the first one and two-item sequence. It is worth noting that the third sequence is redundant as only two plannings are needed to solve the stated task. The sequence was reduced to a two-item sequence $(\mathcal{S}_3, \mathcal{S}_1)$ with another energy gain.

7. Conclusions

In this paper, a class of strongly controllable stratified nonholonomic systems was introduced. For such systems STLC condition is satisfied on each stratum. A motion planner was proposed which exploits the condition not only to reach a goal configuration but also to satisfy additional requirements. It enables the coupling of local and global methods of motion planning on strata to use their advantages. Moreover, it allows the preservation $\mathbb{C}^k$ continuity of the resulting controls only if the controls are represented in a parametric form (possibly different on various strata). The planner was tested on two models of a free-floating space manipulator to check its modules and to draw some remarks. It appears that a short sequences of strata are not necessarily more energetically effective than more numerous ones. It was observed that entry configurations to subsequent strata highly influence the energetic efficiency of motion as the configurations impact the parameters of models for consecutive plannings. Also, additional restrictions on the continuity of controls increase the energy consumption as the controls have to start with the desired values of their derivatives.

References

1. S. M. LaValle, *Planning Algorithms* (Cambridge University Press, Cambridge, UK, 2006).
2. J. Goodwine, Control of Stratified Systems with Robotic Applications *Ph.D. Thesis* (Pasadena, California, USA: California Institute of Technology, 1998).
3. I. Harmati, Stratified Motion and Manipulation Planning Algorithms in Robotics *Ph.D. Thesis* (Budapest, Hungary: Budapest University of Technology and Economics, 2004).
4. I. Harmati, B. Kiss and E. Szadeczky-Kardoss, "On Drift Neutralization of Stratified Systems," In: *Robot Motion and Control*, vol. 335 of *Lecture Notes in Control and Information Sciences* (Springer London, London, 2006) pp. 85–96.
5. W. Chow, "Über systeme von linearen partiellen Differentialgleichungen erster Ordnung," *Math. Ann.* **1**(117), 98–105 (1939).
6. S. Dubowsky and E. Papadopoulos, "The kinematics, dynamics, and control of free-flying and free-floating space robotic system," *IEEE Trans. Robot. Autom.* **9**(5), 531–542 (1993).
7. A. Ellery, "Robotic in-orbit servicers — the need for control moment gyroscopes for attitude control," *Aeronaut. J.* **108**(1082), 207–214 (2004).
8. R. Froelich and H. Papapoff, "Reaction wheel attitude control for space vehicles," *IRE Trans. Autom. Control* **4**(3), 139–149 (1959).
9. A. Ellery, *An Introduction to Space Robotics* (Springer-Verlag, New York, Inc., Secaucus, NJ, USA, 2000).

10. M. Opalka, Motion Planning of Stratified Systems *Ph.D. Thesis* (Wroclaw: Wroclaw University of Technology, 2014).
11. I. Duleba, *Algorithms of Motion Planning for Nonholonomic Robots* (Wroclaw University of Tech. Publishing House, Wrocław, 1998).
12. P. Rouchon, M. Fliess, J. Levine and P. Martin, "Flatness, Motion Planning and Trailer Systems," *Proceeding of the IEEE Conference on Decision and Control*, vol. 3, San Antonio, TX (1993) pp. 2700–2705.
13. G. Lafferriere and H. Sussmann, "Motion Planning for Controllable Systems without Drift," *Proceeding of the IEEE International Conference on Robotics and Automation*, vol. 2, Sacramento, CA (1991) pp. 1148–1153.
14. R. Murray and S. Sastry, "Nonholonomic motion planning: steering using sinusoids," *IEEE Trans. Autom. Contol* **38**(5), 700–716 (1993).
15. K. Tchon and J. Jakubiak, "Endogenous configuration space approach to mobile manipulators: A derivation and performance assessment of jacobian inverse kinematics algorithms," *Int. J. Control* **26**(14), 1387–1419 (2003).
16. Y. Nakamura, *Advanced Robotics: Redundancy and Optimization* (Addison Wesley, New York, 1991).