Acta Cybernetica 15 (2001) 225–240.

Motion Planning Algorithms for Stratified Kinematic Systems with Application to the Hexapod Robot*

István Harmati† and Bálint Kiss†

Abstract

The paper addresses the motion planning problem of legged robots. Kinematic models of these robots are stratified, i.e. the equations of motion differ on different strata. An improved version of the motion planning algorithm proposed in the literature is compared with two alternative solutions via the example of the six-legged (hexapod) robot. The first alternative solution uses explicit integration of the vector fields while the second one exploits the flatness of a restricted subsystem.

1 Introduction

This paper addresses the motion planning problem (MPP) of kinematic systems with stratified configuration spaces [5].

Kinematic systems arise in the presence of non-holonomic constraints [8, 11]. A stratum is a submanifold of the configuration space. Strata may intersect each other leading possibly to a new stratum. We call the bottom stratum the submanifold with the lowest dimension (highest number of constraints). For stratified systems, the equations of motion differ for each submanifold and change discontinuously.

The problem arising in the control of such systems is that the bottom stratum is not controllable (in the sense that the Lie Algebra Rank Condition (LARC) [12, p. 367] is not satisfied). Therefore, one has to switch to other (higher dimensional) strata to find feasible trajectory between two points of the bottom stratum. Successive or cyclic switching between strata is called gaiting.

Since conventional motion planning algorithms (MPA) suppose smooth configuration spaces, they must be modified to work on stratified systems.

*Research partially supported by the Hungarian National Research Program under grant OTKA T 029072 and FKFP 0417/1997

†Department of Control Engineering and Information Technology, Budapest University of Technology and Economics, Pázmány Péter sétany 1/D, Budapest, H-1117 Hungary, Tel: (+361) 4632699, Fax: (+361) 4632204, e-mail: {harmati,bkiss}@seeger.iit.bme.hu

A general MPA, proposed by Lafferriere et al. [9] is adapted in [5, 6, 7] to solve the MPP for stratified systems. This algorithm uses piecewise constant inputs but is imprecise if the Lie algebra generated by the vector fields of the system fails to be nilpotent.

Motion planning can be easily solved for a restricted class of smooth kinematic systems which are differentially flat [3, 4, 10]. For such systems, the MPP is reduced to a simple interpolation problem in the space of the flat output since there is a one-to-one correspondence between sufficiently smooth trajectories of the flat output and feasible trajectories of the system.

Legged robotic structures are typical examples of stratified systems where the possible leg contact combinations define strata in the configuration space. The MPAs presented in the paper will be illustrated using the six-legged robot example where gaiting appears naturally during walking. The first algorithm is an improved version of the method of Lafferriere and Goodwine, the second one is a similar method but allows exact reaching of the final point and easy geometric management of the shape of the trajectory. The third algorithm uses the notion of flatness. The first algorithm is generally applicable while the second and third ones exploit the specific properties of the model and give better results.

The remaining part of the paper is organized as follows. The next section reviews some definitions used in the sequel. Section 3 presents the six-legged robot example in details. Generic methods known from the literature are briefly recalled and illustrated on the hexapod robot example in Section 4. We propose two alternative solutions for the MPP problem related to our example system in Section 5. Some concluding remarks close the paper. Simulation results illustrate all methods.

2 Definitions

The terminology of the differential geometry is used, the reader may refer to [1, 2, 13]. See [3, 4, 10] concerning the notion of differential flatness. We denote by ϕ_g^t the flow [1, p. 238] along the vector field g and by TM the tangent bundle [1, p. 159] of the manifold M.

Definition 1 (kinematic or driftless system). *A kinematic (or driftless) system is a dynamical system of the form:*

$$\dot{x} = \sum_{i=1}^{m} g_i(x)u_i \tag{1}$$

where $x \in \mathbb{R}^n$ and g_i $(i = 1, \ldots, m)$ are real analytic vector fields on $\mathbb{R}^n$.

Let $\mathcal{L}$ denote the Lie algebra generated by the vector fields $g_1, \ldots, g_m$. The system given by (1) is nilpotent with order k if $\mathcal{L}$ is nilpotent with the same order, i.e. $[v_1, [v_2, \ldots, [v_{k-1}, [v_k, v_{k+1}]] \ldots]]$ vanish for $v_i \in \{g_1, \ldots, g_m\}$, $i = 1, \ldots, k+1$.

Definition 2 (point to point MPP). *Consider the system given by (1). The point to point steering MPP consists of finding an input trajectory*

$t \rightarrow (u_1(t), \dots, u_m(t))$ that steers (1) from a given initial state x_I to a desired final state x_F.

A MPA is a systematic procedure that provides an input trajectory (if it exists) solving the MPP for any given pair $(x_I, x_F) \in \mathbb{R}^n \times \mathbb{R}^n$.

Definition 3 (regularly stratified set). *A set $\aleph \subset \mathbb{R}^n$ defined by the union of smooth submanifolds of $\mathbb{R}^n$ (i.e. strata) is said to be a regularly stratified set.*

A stratum S_i is called lower (resp. higher) w.r.t. the stratum S_j if its dimension is lower (resp. higher). The stratum with the lowest dimension is the bottom stratum.

Definition 4 (moving on and moving off vector fields). *Let $\aleph$ be a regularly stratified set. Let S_j and S_i be two strata of $\aleph$ such that S_i is a submanifold of S_j. Let $g \in TS_j$, a vector field on S_j. If $g|S_i \in TS_i$ then g is a moving on vector field of S_i, otherwise g is a moving off vector field of S_i.*

Switching between strata is possible due to the existence of moving off vector fields. A moving on vector field is in general a moving off vector field for lower dimensional substrata.

Definition 5 (stratified kinematic system). *Let $\aleph = \bigcup_{i=1}^{s} S_i$, $\aleph \subset \mathbb{R}^n$ be a regularly stratified set where S_i are smooth submanifolds of $\mathbb{R}^n$. A stratified kinematic system with $\aleph$ as configuration space is given by a set of equations of motion, different on each stratum:*

$$x \in S_i \quad \Rightarrow \quad \dot{x} = \sum_{j \in I_i} g_{i,j}(x) u_j \qquad I_i \subset \{1, \dots, m\} \qquad i = 1, \dots s, \tag{2}$$

where I_i is the set of the active inputs on the stratum S_i and such that the following are satisfied

i. $g_{i,j}$ are moving on vector fields of S_i, i.e. $g_{i,j} \subset TS_i$

ii. $g_{l,j}, g_{k,j} \in TS_l \bigcap TS_k$ for all $x \in S_l \bigcap S_k$ and all $j \in I_k \bigcap I_l$

iii. If the system is on the strata S_i (i.e. $x \in S_i$) at time t and $j \notin I_i$ then $u_j(t) = 0$. This means that the inputs remain always compatible with the stratification of the configuration space.

More than one set of equations of motion are defined for any point x of the configuration space belonging to several strata. In this case the equation of motion is determined by the highest dimensional stratum whose tangent space contains the corresponding $\dot{x}$ and has moving off vector field w.r.t. to all lower dimensional substrata.

Definition 6 (flatness). *The system (1) is differentially flat if one can find a set of variables (flat output)*

$$y = h(x, u, \dot{u}, \ddot{u}, \dots, u^{(r)}), \qquad y \in \mathbb{R}^m$$

with r finite integer, such that

$$x = \alpha(y, \dot{y}, \ddot{y}, \dots, y^{(q)}) \qquad u_i = \beta_i(y, \dot{y}, \ddot{y}, \dots, y^{(q+1)}) \qquad i = 1, \dots, m$$

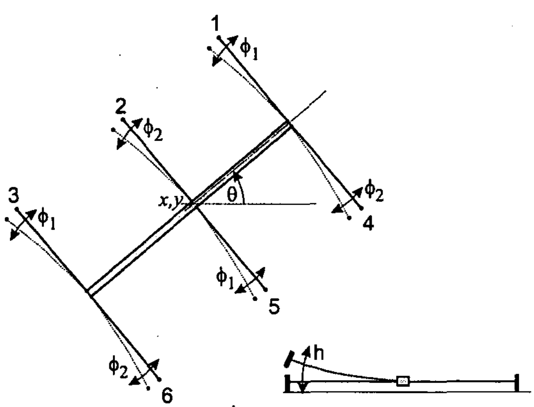
with q a finite integer, and such that the system equations

$$\frac{d\alpha}{dt}(y, \dot{y}, \ddot{y}, \dots, y^{(q+1)}) = \sum_{j=1}^{m} g_j(\alpha(y, \dot{y}, \ddot{y}, \dots, y^{(q)})) \cdot \beta_j(y, \dot{y}, \ddot{y}, \dots, y^{(q+1)})$$

are identically satisfied.

3 The hexapod robot example

Legged robots are stratified systems since their kinematics change discontinuously when a leg makes or brakes contact with the ground. The discussed motion planning methods are illustrated on the example of the hexapod robot (introduced in [5]), depicted in Figure 1 (left). The position and orientation of the body is given by



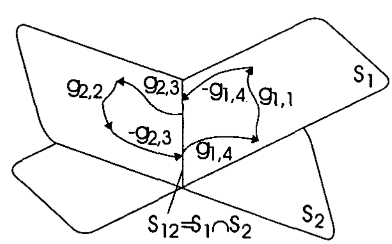


Figure 1: The hexapod robot (left); A tripod gait in the stratified configuration space of the hexapod robot (right)

the variables x, y and θ. The legs $\{1,3,5\}$ and the legs $\{2,4,6\}$ have the same leg angles (ϕ_1 and ϕ_2) and move (touch the ground and brake contact) simultaneously. The hexapod robot is assumed to be in stable position if the leg angles do not leave an admissible range given by $[\phi_{min}, \phi_{max}]$. The distance of the two set of legs from the ground is given by h_1 and h_2, respectively. Thus the configuration space is a regularly stratified set $\aleph$ in $M = \mathbb{R}^2 \times \mathbb{S}^3 \times \mathbb{R}^2$, where $\mathbb{S}$ is the unit circle. The state vector is given by $\xi = (x, y, \theta, \phi_1, \phi_2, h_1, h_2)^T$. We have the following strata:

$$\begin{array}{ll} S_1 = \{\xi \in M : h_1 = 0\}, I_1 = \{1,2,4\}, u_3 = 0, & \text{legs } \{1,3,5\} \text{ on the ground} \\ S_2 = \{\xi \in M : h_2 = 0\}, I_2 = \{1,2,3\}, u_4 = 0, & \text{legs } \{2,4,6\} \text{ on the ground} \\ S_{12} = \{\xi \in M : h_1 = h_2 = 0\}, I_{12} = \{1,2\}, u_3 = u_4 = 0, & \text{all legs on the ground} \end{array}$$

The robot is able to perform different kind of motions:

1. If only one of the set of legs $\{1,3,5\}$ (resp. $\{2,4,6\}$) touches the ground (active legs) while the other set of legs remains in the air (inactive legs) then the body may move by changing the angles ϕ_1 (resp. ϕ_2). In this case, only the active legs have influence on the planar motion of the robot, but the inactive legs may also change their leg angles. The successive commutation of the active and inactive legs (called gaiting) makes the walking possible.
2. If all legs touch the ground at the same time then all legs are active. The angles ϕ_1 and ϕ_2 may change. This motion will be referred to as the paddling motion.

These motions are possible along the vector fields:

$$\begin{aligned} g_{12,1}(\xi) &= \begin{pmatrix} \cos\theta & \sin\theta & l & 1 & 0 & 0 & 0 \end{pmatrix}^T \\ g_{12,2}(\xi) &= \begin{pmatrix} \cos\theta & \sin\theta & -l & 0 & 1 & 0 & 0 \end{pmatrix}^T \\ g_{1,1}(\xi) &= g_{12,1}(\xi) \\ g_{1,2}(\xi) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}^T \\ g_{1,4}(\xi) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}^T \\ g_{2,1}(\xi) &= \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}^T \\ g_{2,2}(\xi) &= g_{12,2}(\xi) \\ g_{2,3}(\xi) &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T \end{aligned} \tag{3}$$

where l denotes the length of the legs ($l = 0.1$m is used for simulations). The equations of motion in the different strata read:

$$\xi \in S_1 \Rightarrow \quad \dot{\xi} = g_{1,1}(\xi)u_1 + g_{1,2}(\xi)u_2 + g_{1,4}(\xi)u_4 \tag{4a}$$

$$\xi \in S_2 \Rightarrow \quad \dot{\xi} = g_{2,1}(\xi)u_1 + g_{2,2}(\xi)u_2 + g_{2,3}(\xi)u_3 \tag{4b}$$

$$\xi \in S_{12} \Rightarrow \quad \dot{\xi} = g_{12,1}(\xi)u_1 + g_{12,2}(\xi)u_2 \tag{4c}$$

Thus the moving on vector fields of the (bottom) stratum S_{12} are $g_{12,1}$ and $g_{12,2}$. Since $S_{12} = S_1 \bigcap S_2$, the moving on vector fields $g_{1,4}$ and $g_{2,3}$ of S_1 and S_2, respectively, are moving off vector fields of S_{12} (lift legs $\{1,3,5\}$ or legs $\{2,4,6\}$).

Remark 1. *Since the leg heights h_1 and h_2 do not change along the integral curves of the vector fields $g_{1,2}$ and $g_{2,1}$ (the last two components of these vector fields vanish) we have that $g_{1,2}(\xi) \in TS_{12}$ and $g_{2,1}(\xi) \in TS_{12}$ for all $\xi \in S_{12}$, although $g_{1,2}$ and $g_{2,1}$ are defined on S_1 and S_2, respectively.*

Definition 7 (tripod gait). *A flow sequence of the form*

$$\xi_F = \underbrace{\phi^{t_8}_{-g_{2,3}}}_{S_{12}\leftarrow S_2} \circ \underbrace{\phi^{t_7}_{g_{2,2}}}_{on\ S_2} \circ \underbrace{\phi^{t_6}_{g_{2,1}}}_{on\ S_2} \circ \underbrace{\phi^{t_5}_{g_{2,3}}}_{S_2\leftarrow S_{12}} \circ \underbrace{\phi^{t_4}_{-g_{1,4}}}_{S_{12}\leftarrow S_1} \circ \underbrace{\phi^{t_3}_{g_{1,2}}}_{on\ S_1} \circ \underbrace{\phi^{t_2}_{g_{1,1}}}_{on\ S_1} \circ \underbrace{\phi^{t_1}_{g_{1,4}}}_{S_1\leftarrow S_{12}} (\xi_I), \qquad (5)$$

such that $\xi_F, \xi_I \in S_{12}$ is called a tripod gait.

The stratification of the configuration space of the hexapod robot with the flow during a tripod gait is illustrated in Figure 1 (right).

The MPP related to the hexapod robot consists of finding control inputs $t \to u_i(t)$, $i = 1, \ldots, 4$ such that the generated trajectory connects the points p and q in the *bottom stratum* S_{12} with $p = (x_I, y_I, \theta_I, \phi_{1,I}, \phi_{2,I})^T$ to $q = (x_F, y_F, \theta_F, \phi_{1,F}, \phi_{2,F})^T$.

4 Generic methods

4.1 An MPA for smooth kinematic systems

Based on the paper of Lafferriere and Sussmann [9] we present first a motion planning algorithm for the smooth kinematic system (1) which will be extended for stratified systems in the following subsection using [5]. We assume that (1) is completely controllable [12, p. 367]. Suppose that we want to find a trajectory connecting p to q. The algorithm proposed in [9] is the following.

Algorithm 1:

Step I. Extend the system (1) to

$$\dot{x} = v_1 g_1(x) + \ldots + v_m g_m(x) + v_{m+1} g_{m+1}(x) + \ldots + v_r g_r(x) \qquad (6)$$

where the vector fields $g_{m+1}, \ldots, g_r$ ($r \geq n$) are defined by higher order Lie brackets of g_i, $i = 1, \ldots, m$, selected such that $\text{span}\{g_1(x), \ldots, g_r(x)\} = \mathbb{R}^n$ (This is always possible due to the controllability assumption.)

Step II. Find a control v that steers the extended system (6) from p to q. Since the vector fields $\{g_1(x), \ldots, g_r(x)\}$ span $\mathbb{R}^n$, one can choose a straight line segment between p and q. The corresponding input v of (6) is obtained by matrix inversion.

Step III. We compute a control u for the original system (1) that substitutes the control v of the extended system (6). The substitution means that the trajectory of (1) obtained by applying u connects the same initial and final points p and q, as the straight line trajectory of the extended system (6) under the action of v. This is done using the following steps:

Step 1. We calculate the order of nilpotency of the Lie algebra $\mathcal{L}$ associated to system (1). If the order of nilpotency is not finite, then we use a kth order finite approximation of the Lie algebra $\mathcal{L}$ by replacing the brackets $[v_1, [v_2, \ldots, [v_{k-1}, [v_k, v_{k+1}]] \ldots]]$ by zero for $v_i \in \{g_1, \ldots, g_m\}$, $i = 1, \ldots, k+1$.

Step 2. We determine the Philip Hall basis of $\mathcal{L}$ [9, § 6], [11, p. 704]. The elements of this basis are symbolic Lie brackets of the vector fields $g_1, \dots, g_m$.

Step 3. We solve the formal differential equation [12]

$$\dot{S}(t) = S(t) \left(\sum_{j=1}^{r} v_j g_j \right) \qquad S(0) = 1, \tag{7}$$

where the solution $S(t)$ is an element of a special nilpotent Lie group [9, § 7] and has the form

$$S(t) = e^{\tilde{h}_1(t)B_1} \dots e^{\tilde{h}_{s-1}(t)B_{s-1}} e^{\tilde{h}_s(t)B_s}, \tag{8}$$

where B_i are the elements of the P. Hall basis and $\tilde{h}_i(t)$ are the forward P. Hall coordinates. The symbolic expression of $S(t)$ gives a sequence of flows along the vector fields of the P. Hall basis of $\mathcal{L}$.

Step 4. Let $q = S(1) \cdot p$. Introduce $\bar{h}_i = \tilde{h}(1)$ for $(i = 1, \dots, r)$. Then we get

$$q \approx e^{\bar{h}_1 B_1} \dots e^{\bar{h}_{s-1} B_{s-1}} e^{\bar{h}_s B_s} \cdot p. \tag{9}$$

where we have equality if no nilpotent approximation is needed.

Step 5. Due to the nilpotency of $\mathcal{L}$ (or to its nilpotent approximation, see Step 1), the Campbell-Baker-Hausdorff formula [13, p. 114] allows to replace the flows along basis elements obtained by Lie brackets with a sequence of flows along the vector fields $g_1, \dots, g_m$. This gives

$$q \approx \left(\prod_{i=1}^{l<\infty} e^{\chi_i X_i} \right) p \qquad X_i \in \{g_1, \dots, g_m\} \tag{10}$$

where χ_i are constant.

Step 6. Using (10), the piecewise constant input of the original system in the time interval $i - 1 < t < i$, $i = 1, \dots, l)$ is given by

$$\begin{array}{lll} u_j(t) = 0 & \text{if} & X_i \neq g_j \\ u_j(t) = \chi_i & \text{if} & X_i = g_j \end{array} \qquad j = 1, \dots, m. \tag{11}$$

Note that the trajectories along which the original system (1) and the extended system (6) arrive to q are, in general, different.

Remark 2. *The motion planning algorithm gives trajectories such that only one of the inputs is nonzero at the same time. The exact shape of the trajectory can not be planned explicitly.*

Remark 3. *In practice, if the order of nilpotency is too high for numeric calculations, then one may use again a lower order approximation forcing the high order brackets to vanish similarly to the case of non-nilpotent systems.*

4.2 Application to stratified systems

By Definition 5, different strata give different equations of the type (1). Since the controllability on a given stratum is not guaranteed, Algorithm 1 can not be directly applied. To overcome this difficulty, vector fields from higher strata are also considered.

Detailed presentation can be found in [5, 6, 7]. Let us directly turn our attention to the hexapod robot example. Note that the system is not controllable on S_{12} because the Lie algebra generated by the moving on vector fields ($g_{12,1}$ and $g_{12,2}$) of the bottom stratum does not satisfy the LARC.

Lemma 1. *For any sequence of flows along the vector fields $g_{12,1}$, $g_{12,2}$, $g_{2,1}$, and $g_{1,2}$ in the bottom stratum S_{12}, connecting p to q, there exists a sequence of tripod gaits connecting p to q.*

Proof. Without loss of generality, consider the following flow sequence in the bottom stratum:

$$\xi_F = \phi^{t_4}_{g_{12,2}} \circ \phi^{t_3}_{g_{2,1}} \circ \phi^{t_2}_{g_{1,2}} \circ \phi^{t_1}_{g_{12,1}}(\xi_I) \tag{12}$$

Using (3), replace $\phi^{t_4}_{g_{12,2}}$ by $\phi^{t_4}_{g_{2,2}}$ and $\phi^{t_1}_{g_{12,1}}$ by $\phi^{t_1}_{g_{1,1}}$:

$$\xi_F = \phi^{t_4}_{g_{2,2}} \circ \phi^{t_3}_{g_{2,1}} \circ \phi^{t_2}_{g_{1,2}} \circ \phi^{t_1}_{g_{1,1}}(\xi_I). \tag{13}$$

Insert the flows sequences $\phi^{t_5}_{-g_{2,3}} \circ \phi^{t_5}_{g_{2,3}} = I$ and $\phi^{t_6}_{-g_{1,4}} \circ \phi^{t_6}_{g_{1,4}} = I$:

$$\xi_F = \phi^{t_5}_{-g_{2,3}} \circ \phi^{t_5}_{g_{2,3}} \circ \phi^{t_4}_{g_{2,2}} \circ \phi^{t_3}_{g_{2,1}} \circ \phi^{t_2}_{g_{1,2}} \circ \phi^{t_1}_{g_{1,1}} \circ \phi^{t_6}_{-g_{1,4}} \circ \phi^{t_6}_{g_{1,4}}(\xi_I). \tag{14}$$

Since $[g_{1,4}, g_{1,1}] = 0$, $[g_{1,4}, g_{1,2}] = 0$, $[g_{2,3}, g_{2,1}] = 0$ and $[g_{2,3}, g_{2,2}] = 0$, the flows along these pairs of vector fields commute. This allows to rearrange the terms to obtain:

$$\xi_F = \phi^{t_5}_{-g_{2,3}} \circ \phi^{t_4}_{g_{2,2}} \circ \phi^{t_3}_{g_{2,1}} \circ \phi^{t_5}_{g_{2,3}} \circ \phi^{t_6}_{-g_{1,4}} \circ \phi^{t_2}_{g_{1,2}} \circ \phi^{t_1}_{g_{1,1}} \circ \phi^{t_6}_{g_{1,4}}(\xi_I), \tag{15}$$

which is a tripod gait by Definition 7. □

Based on the previous lemma and Algorithm 1, the following algorithm is proposed to solve the MPP of the hexapod robot being a stratified kinematic system [5, 6].

Algorithm 2:

Step I. Construct a smooth kinematic system on S_{12} from the vector fields $g_{12,1}$, $g_{12,2}$, $g_{1,2}$ and $g_{2,1}$ referred to as the bottom stratified system:

$$\dot{\xi} = g_{12,1}(\xi)w_1 + g_{12,2}(\xi)w_2 + g_{1,2}(\xi)w_3 + g_{2,1}(\xi)w_4 \quad \xi \in S_{12}, \tag{16}$$

where $w_1, \dots, w_4$ is the input vector. Note that this system is controllable in S_{12}.

Step II. Use Algorithm 1 on the bottom stratified system (16), to obtain the piecewise constant input sequence which makes possible to connect p to q. Since the Lie algebra associated to (16) fails to be nilpotent, a nilpotent approximation has to be used. The solution, which reaches a point $\tilde{q}$ close to q, gives a sequence of flows along the vector fields $g_{12,1}$, $g_{12,2}$, $g_{1,2}$ and $g_{2,1}$:

$$q \approx \tilde{q} = \phi^{\chi_s}_{X_s} \circ \phi^{\chi_{s-1}}_{X_{s-1}} \circ \cdots \phi^{\chi_1}_{X_1}(p) \quad X_i \in \{g_{12,1}, g_{12,2}, g_{1,2}, g_{2,1}\}. \tag{17}$$

Step III. Using Lemma 1, one can find a gaiting sequence by inserting flows of moving off vector fields of the strata S_{12} where the successive flows are in different strata:

$$q \approx \tilde{q} = \phi^{\chi_l}_{X_l} \circ \cdots \phi^{\chi_1}_{X_1}(p) \quad X_i \in \{g_{12,1}, g_{12,2}, g_{1,1}, g_{1,2}, g_{1,4}, g_{2,1}, g_{2,2}, g_{2,3}\}, \tag{18}$$

with $l > s$ due to the insertions.

Step IV. Let T be the desired travelling time between p and q. Introduce $T_s = \sum_{i=1}^{l} \chi_i$. In order to arrive to $\tilde{q}$ at time T, the inputs and the switching time are obtained from (18) as

$$\begin{aligned} u_1(t) &= \begin{cases} \frac{T_s}{T}\chi_i & \text{if } X_i \in \{g_{12,1}, g_{1,1}, g_{2,1}\} \\ 0 & \text{otherwise} \end{cases} & u_3(t) &= \begin{cases} \frac{T_s}{T}\chi_i & \text{if } X_i = g_{2,3} \\ 0 & \text{otherwise} \end{cases} \\ u_2(t) &= \begin{cases} \frac{T_s}{T}\chi_i & \text{if } X_i \in \{g_{12,2}, g_{1,2}, g_{2,2}\} \\ 0 & \text{otherwise} \end{cases} & u_4(t) &= \begin{cases} \frac{T_s}{T}\chi_i & \text{if } X_i = g_{1,4} \\ 0 & \text{otherwise,} \end{cases} \end{aligned} \tag{19}$$

for the time interval $(i-1)\frac{T}{T_s} < t < i\frac{T}{T_s}$, $i = 1, \ldots, l$.

If the bottom stratified system is not nilpotent, then the approximation error can be decreased by inserting intermediate reference points between p and q. The section between two neighboring intermediate reference points is called a subsegment. Then the above algorithm can be applied successively for the subsegments. The major problem of the method is the choice of the length of the subsegments along the trajectory. There exists a critical distance under which the algorithm converges (see [9]), however its estimation is a hard question without theoretical answer. The proposal of [6, 7] can be used modifying iteratively the appropriate length during the computation.

4.3 Software implementation and simulation

Let us illustrate the preceding stratified MPA (Algorithm 2) with a simulation software[1] written using the improvements of [6, 7], which can be also exploited for more complex problems (e.g. dextrous manipulation with robotic hands). The simulation results for the hexapod robot with two different prescribed orientations are

[1] *All simulation programs are written using Matlab and the Symbolic Math Toolbox.*

given in Figure 2 (left: constant orientation, right: tangent orientation). Figure 2 shows the reference points and the developed path of the hexapod robot, the orientation is not illustrated. The x, y coordinate axes are scaled in m. The trajectories are specified using the same sequence of reference points in the x, y plane but the orientations at the reference points are different. It can be seen that in spite of the

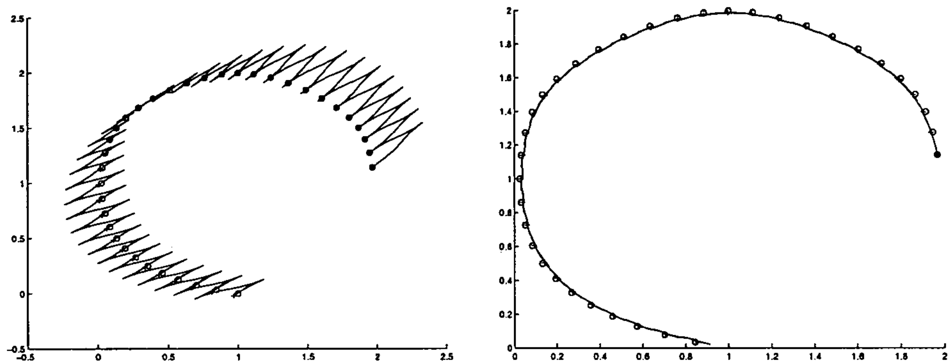


Figure 2: Stratified motion planning for hexapod robot: *i*) constant orientation, $\theta = \frac{\pi}{3}$radian (left); *ii*) tangent orientation; "o": reference points, "+": reached points (right)

fact that the Lie algebra $\mathcal{L}$ of the vector fields of (3) fails to be nilpotent, the solution based on a nilpotent approximation of order 1 gives satisfactory accuracy for both cases. If the desired orientation is close to the tangent of the curve connecting the reference points, then the algorithm has less difficulty to provide more precise solution.

5 Two alternative MPP solutions

5.1 Exact reaching along circle arcs with piecewise constant inputs

Assume again, as proposed in [9], that the inputs are piecewise constant and all but one input vanish at the same time.

First, observe that the flows along the vector fields $g_{12,1}$ (resp. $g_{12,2}$) are circles in the x, y plane. (Recall that in S_1 (resp. in S_2) the input u_3 (resp. u_4) vanish.) To see this, integrate $g_{12,1}$ with the initial conditions $\theta(0) = \theta_0$, $x(0) = x_0$, $y(0) = y_0$

and with $u_1(t) = \pm u_0$ (with $u_0 > 0$ and constant). We obtain

$$\begin{aligned}\theta(t) &= \pm l u_0 t + \theta_0 \\ y(t) &= -\frac{1}{l}\cos(\pm l u_0 t + \theta_0) + \frac{1}{l}\cos\theta_0 + y_0 \\ x(t) &= \frac{1}{l}\sin(\pm l u_0 t + \theta_0) - \frac{1}{l}\sin\theta_0 + x_0,\end{aligned}$$

which leads to

$$\left(y(t) - \frac{1}{l}\cos\theta_0 - y_0\right)^2 + \left(x(t) + \frac{1}{l}\sin\theta_0 - x_0\right)^2 = \frac{1}{l^2},$$

giving a circle. Integrating $g_{12,2}$ with constant input $u_2(t) = \pm u_0$ gives similar result. Consequently, if we restrict ourselves to trajectories along the flows $g_{12,1}$ and $g_{12,2}$ with constant velocities, the robot may follow arcs of radius $\frac{1}{l}$. Given a position and an orientation (x, y, θ) of the hexapod robot in the plane, an integral curve of the vector field $g_{12,1}$ passing through (x, y, θ) is referred to as an arc of type 1 and an integral curve of the vector field $g_{12,2}$ as an arc of type 2.

Definition 8 (Admissible trajectory). *A sequence of arcs in the plane is said to be an admissible trajectory for the hexapod robot if each two subsequent arcs, a_i and a_{i+1} have a unique common point where the tangent directions to both arcs coincide and, additionally, a_i and a_{i+1} are of different type.*

A simple trajectory planning method that constructs an admissible trajectory between two points $p = (x_I, y_I, \theta_I, \phi_{1,I}, \phi_{2,I})^T$ and $q = (x_F, y_F, \theta_F, \phi_{1,F}, \phi_{2,F})^T$ is based on the construction of tangent circles.

Algorithm 3:

Step I. Determine the flows (circles in the x, y plane) of type 1 and 2 passing through the point x_I, y_I and choose the one with the origin closer to x_F, y_F. If the distances of the origins of the two circles from the point x_F, y_F are the same, chose the circle of type one. Denote this circle by C_I.

Step II. Determine the flows (circles in the x, y plane) of type 1 and 2 passing through the point x_F, y_F and choose the one with the origin closer to the that of C_I. If the distances of the origins of the two circles from that of C_I are the same, chose the circle of type one. Denote this circle by C_F.

Step III. Calculate the unit vector

$$v = \frac{1}{\sqrt{(x_F - x_I)^2 + (y_F - y_I)^2}} \begin{bmatrix} x_F - x_I \\ y_F - y_F \end{bmatrix}$$

and let $\mathcal{C} = \{C_I\}$ a list of circles with unique element C_I.

Step IV. Get the last circle from list $\mathcal{C}$ and denote its origin by (x_C, y_C) and its type by $h \in \{1, 2\}$. Append to the list $\mathcal{C}$ a circle of radius $\frac{1}{l}$ with origin

$$\begin{bmatrix} x_C \\ y_C \end{bmatrix} + \frac{2}{l} v$$

and let the type of this circle be different of h. Repeat this step until *both* of the following conditions are fulfilled:

- the distance of the origin of the newly appended circle from the origin of C_F is less or equal to $\frac{4}{l}$,
- the type of the newly appended circle is the same than that of C_F.

Step V. Construct a circle which is tangent to the last circle of the list $\mathcal{C}$ and to C_F and let its type be different from that of C_F. Append this circle to the list $\mathcal{C}$. Append C_F to the list $\mathcal{C}$.

Step VI. The reference trajectory is given by the sequence of arcs of the tangent circles of the list $\mathcal{C}$.

Step VII. The contact points between the tangent circles allow to calculate the change of orientation (the evolution of θ) along each arc.

Step VIII. Since the input is constant along each arc the travelling time belonging to every arc can be obtained from the equation $\dot{\theta}(t) = lu_1(t) - lu_2(t)$. The travelling time T_i along the ith circle arc can be expressed from

$$\frac{\theta_i - \theta_{i-1}}{T_i} = lu_1(t) - lu_2(t). \tag{20}$$

Step IX. The travelling time along the arcs allow to calculate the variation of the leg angles ϕ_i, $i = 1, 2$, and thus the moments where ϕ_i leaves its admissible range $[\phi_{min}, \phi_{max}]$. These moments specify the gaits when the corresponding legs are needed to be lift off and rotated back by insertion of moving off vector fields ($g_{1,4}$ and $g_{2,3}$ to lift off and put down the legs).

Note that the recurrence defined in **Step IV** always terminates after a finite number of iterations if the distance of the initial and the final point is finite. Note also that the construction of the circle in **Step V** is always possible since the distance of the origins of the last circle of $\mathcal{C}$ and C_F is less than $\frac{4}{l}$.

The algorithm guarantees the exact reach of the final point. The shape of the trajectory is in direct control (because it consists of simple curve pieces, i.e. arcs) and this makes possible to extend the algorithm with obstacle avoidance.

The path constructed by the algorithm is optimal neither in length nor in number of gaits. In fact, by appending shorter sections of arcs, the straight line trajectory, optimal in length, can be more closely approximated. Moreover, since the legs have to be rotated back periodically anyway, gaits at those moments are not prohibitive and could allow shorter arcs than half-circles inserted by our algorithm. The deeper analysis of the questions of finding paths which are optimal in terms of gaits or length is beyond the scope of the present paper.

Figure 3 gives the trajectory connecting $p = (-50, -10, 0, 0, 0)$ and $q = (5, -5, -\frac{3\pi}{4}, 0, 0)$. The velocity along the flows $g_{12,1}$ and $g_{12,2}$ is $u_i = \pm u_0 = \pm 5$, $i = 1, 2$.

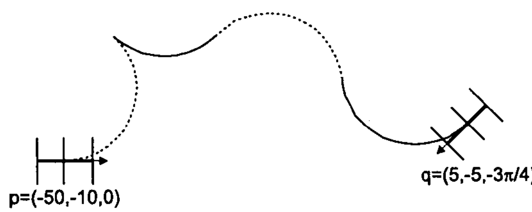


Figure 3: Trajectory along arcs of circles

5.2 Exact reaching along sufficiently smooth curve in the x, y plane using paddling motion

Proposition 1. *The kinematics of the bottom stratum* (4c), *restricted to the tangent space of* (x, y, θ), *and given by the equations*

$$\dot{x} = \cos\theta(u_1 + u_2) \tag{21a}$$

$$\dot{y} = \sin\theta(u_1 + u_2) \tag{21b}$$

$$\dot{\theta} = l(u_1 - u_2), \tag{21c}$$

is differentially flat. A flat output is the position of the robot: (x, y).

Proof. From the first two equations one calculates the trajectory of θ using $\dot{x}$, $\dot{y}$:

$$\theta = \arctan 2(\dot{y}, \dot{x}), \tag{22}$$

where arctan 2 is the inverse function of tan which gets its value in the interval $(-\pi, \pi]$. Then, using (21a) or (21b) together with (21c), one calculates u_1 and u_2 as function of $\dot{x}, \dot{y}, \theta, \dot{\theta}$. But θ is in turn function of $\dot{x}, \dot{y}$. Consequently, all variables are functions of $\dot{x}, \ddot{x}, \dot{y}, \ddot{y}$, the successive time derivatives of x and y which proves that the system is flat with the flat output (x, y) as claimed. □

Flatness implies that the (sufficiently smooth) trajectory of the position determines completely the trajectory of the remaining variables of the restricted model (θ, u_1, and u_2). In the case of the hexapod robot, the desired trajectory in the x, y plane must be continuously differentiable.

The variables ϕ_1 and ϕ_2 are not included in the restricted model (21). However, their trajectory and the gait moments can be obtained by integration of u_1 and u_2 using the relations $\dot{\phi}_1 = u_1$, $\dot{\phi}_2 = u_2$. Their desired final values $(\phi_{1,F}, \phi_{2,F})$ can be reached by lifting off and rotating the corresponding legs which leaves unchanged the position and orientation of the hexapod robot. This gives the following MPA:

Algorithm 4:

Step I. Let the reach time be fixed to T. Construct at least third order polynomials for $x(t)$ and y(t) whose coefficients are determined using the constraints

$$x(0) = x_I \qquad x(T) = x_F \qquad y(0) = y_I \qquad y(T) = y_F$$
$$\dot{x}(0) = \cos\theta_I \qquad \dot{x}(T) = \cos\theta_F \qquad \dot{y}(0) = \sin\theta_I \qquad \dot{y}(T) = \sin\theta_F.$$

Step II. Calculate the trajectory of θ, u_1, and u_2 as a function of $x(t), y(t)$ and their time derivatives. This is possible since the corresponding subsystem is flat.

Step III. Integrate numerically u_i $(i = 1, 2)$ to obtain ϕ_i. Each time when ϕ_i goes out of range insert a gait which lifts off and rotates back the corresponding legs. When the desired position and orientation is reached, lift off the legs and rotate them until the desired final leg angles are reached.

Remark 4. *Choosing T too small results large u_i. If there are limitations on u_i, they can be respected by re-scaling the time along the trajectory (control of the clock), and/or by increasing T.*

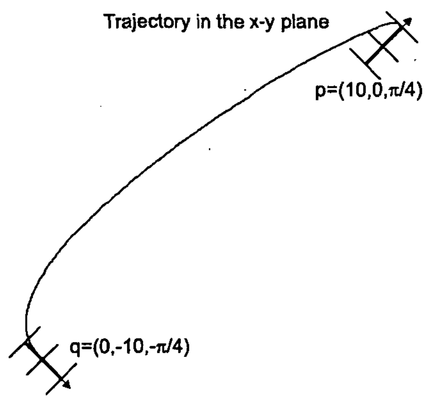


Figure 4: Motion planning along arbitrary curve (paddling motion) using flatness.

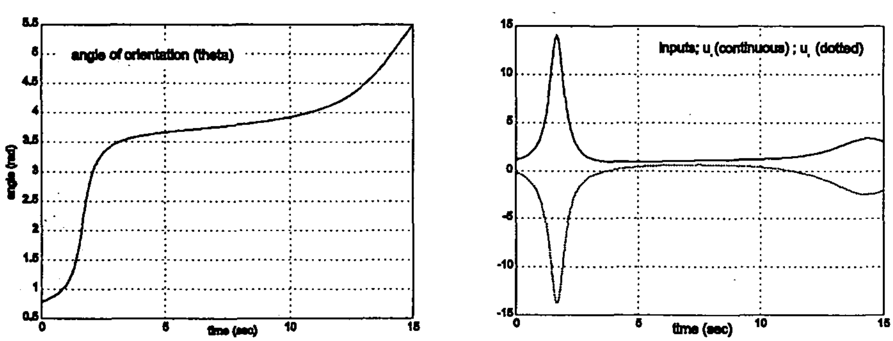


Figure 5: Trajectory of the orientation and the inputs using paddling motion.

A trajectory connecting $p = (10, 0, \pi/4)$ and $q = (0, -10, -\pi/4)$ is illustrated in Figure 4. The evolution of the orientation angle θ and the input trajectory are given in Figure 5.

6 Conclusion

The kinematics of walking robots give rise to stratified systems. We presented three algorithms solving the steering MPP of the hexapod robot. It turns out that the generic methods can be improved if the particular geometry of the given system is exploited which is the case of the last two MPAs. However, these alternative methods (Algorithms 3 and 4) use the specific properties of the robot model, thus they cannot be widely generalized.

The first alternative method presented in Subsection 5.1 using simple geometric manipulation is well adapted to real-time trajectory generation since it involves neither numerical integration nor optimization.

Both proposed alternative methods are able to reach exactly the desired final point. This is not possible (without the use of feedback techniques) for non-nilpotent systems (e.g. the hexapod robot, considered in the paper) using the generic methods. The additional advantages are that the alternative algorithms work precisely without the insertion of additional reference points between the starting and final points.

The motion planning problem discussed in this paper works on a kinematic model of the hexapod robot. The extension to a model incorporating friction effects at the leg contacts, more complicated leg kinematics, the dynamics of the legs and that of the robot itself needs to extend existing MPAs from driftless systems to a more general class of nonlinear control mechanical systems. The problem remains tractable if the extended model remains flat, but is a difficult one for general stratified mechanical systems and is still an open research area.

References

[1] R. Abraham, J. E., Marsden, and T. Ratiu. *Manifolds, tensor analysis and applications.* Springer - Verlag New York Inc., 1988.

[2] D. V. Alekseevskij, A. M. Vinogradov, and Lychagin V. V. *Geometry I - Basic Ideas and Concepts of Differential Geometry*, volume 28 of *Encyclopedia of Mathematical Sciences.* Springer - Verlag New York Inc., 1991.

[3] M. Fliess, J. Lévine, Ph. Martin, and P. Rouchon. Flatness and defect of nonlinear systems: introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

[4] M. Fliess, J. Lévine, Ph. Martin, and P. Rouchon. A Lie Bäcklund approach to equivalence and flatness of nonlinear systems. *IEEE Transactions on Automatic Control*, 38:700–716, 1999.

[5] B. Goodwine. *Control of stratified systems with robotic applications.* PhD thesis, California Institute of Technology, 1998.

[6] I. Harmati, B. Lantos, and S. Payandeh. Extensions of nonlinear control concept to object manipulation with finger relocation. In *Proceedings of the 6th International IFAC Symposium on Robot Control*, pages 223–228, Vienna, Austria, 21-23 September, 2000.

[7] I. Harmati, B. Lantos, and S. Payandeh. Improved stratified control for hexapod robots and object manipulation with finger relocation. *In Press. Periodica Polytechnica, Budapest, Hungary*, 2000.

[8] S. D. Kelly and R. M. Murray. Geometric phases and robotic locomotion. *Journal of Robotic Systems*, 12(6):417–431, 1995.

[9] G. Lafferriere and H. J. Sussmann. Motion planning for controllable systems without drift. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1148–1153, Sacramento, CA, USA, 1991.

[10] Ph. Martin and P. Rouchon. Feedback linearization and driftless systems. *Mathematics of Control Signals, and Systems*, 7(3):235–254, 1994.

[11] R. M. Murray and S. S. Sastry Nonholonomic Motion Plannaing: Steering Using Sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.

[12] H. J. Sussmann, New differential geometric methods in nonholonomic path finding *Systems, Models, and Feedback: Theory and Applications*, A. Isidori and T. J. Tarn Eds, Birkhäuser, Boston, pp. 365-384., 1992

[13] V. Varadarajan, *Lie Groups, Lie Algebras, and Their Representations.* Springer-Verlag, 1994.