Nonlinear Control Systems Design (NOLCOS)
Enschede, the Netherlands, July 1998

MOTION PRIMITIVES FOR STABILIZATION AND CONTROL OF UNDERACTUATED VEHICLES

Francesco Bullo *,[1] **Naomi Ehrich Leonard** **,[2]

* *Control and Dynamical Systems, Mail Stop 107-81, California Institute of Technology, Pasadena, CA 91125*
bullo@cds.caltech.edu
** *Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544*
naomi@princeton.edu

Abstract: In this paper we construct motion primitives as algorithm building blocks for stabilization and control of a class of underactuated systems that includes spacecraft, submersibles and planar vehicles. The underactuated systems are modelled as Lagrangian systems on Lie groups; thus, they are systems with drift and with accessibility distributions described by the operation of Lie bracket and symmetric product. Directions of motion that are not directly actuated are identified with symmetric products of the input vector fields, and in-phase sinusoidal forcing is used in the primitives to generate motion in these directions. These primitives can then be used for a variety of low velocity maneuvers. For example, we demonstrate their use for exponential point stabilization and for static interpolation. We evaluate our algorithms and investigate the advantage of planning motions along relative equilibria.

Keywords: underactuated systems, mechanical systems, exponential stabilization

1. INTRODUCTION

Underactuated mechanical control systems provide a challenging research area of increasing interest in both application and theory. In this paper, we design stabilization and motion control laws for an important class of underactuated mechanical control systems that includes underwater vehicles, satellites, surface vessels, airships and hovercrafts. For these systems, relevant Lagrangian models are available and lift/drag type effects are sometimes negligible. Key features are the following: (1) the configuration space is a Lie group, as, for example, the group of rotations SO(3) in the case of a satellite, (2) the Lagrangian is equal to the kinetic energy, (3) external forces are fixed with respect to the body, and (4) there are fewer actuators than degrees of freedom.

These systems offer a control challenge as they have non-zero drift, their linearization at zero velocity is not controllable and they are generically neither feedback linearizable nor nilpotent. Further, no test is available to establish whether or not they are differentially flat. In other words, the motion planning problem for this class of systems cannot be solved with any established method.

Important references include the work on small-time local controllability (Sussmann, 1987) and on configuration controllability for simple mechanical systems (Lewis and Murray, 1997). Regarding the constructive controllability problem, we employ an analogous approach to one in (Leonard and Krishnaprasad, 1995), where motion algorithms for a class of kinematic systems on Lie groups were

[1] Research supported in part by the National Science Foundation under Grant CMS-9502224.
[2] Research supported in part by the National Science Foundation under Grant BES-9502477 and by the Office of Naval Research under Grant N00014-96-1-0052.

designed with small-amplitude periodic inputs. We achieve exponential stabilization by iterating an approproximate stabilization step in the spirit of (Lafferriere and Sussmann, 1991).

The paper is organized as follows. In Section 2 we review from (Bullo and Leonard, 1997) an algebraic test for local controllability at zero velocity and a perturbation analysis of mechanical systems under small amplitude forcing. In Section 3 we design two motion primitives that perform the basic tasks of changing and maintaining velocity using in-phase inputs. The two motion primitives are the building blocks for designing high-level motion procedures; we design algorithms to solve point-to-point reconfiguration, local exponential stabilization and static interpolation problems. Our algorithms provide suboptimal control laws; however, we show how the interpolation algorithm applied to a path along a sequence of relative equilibria exploits system dynamics to provide an efficient motion control strategy.

ACKNOWLEDGEMENTS

The first author thanks Prof. Richard M. Murray and Dr. Andrew D. Lewis for helpful discussions.

2. MODELS, CONTROLLABILITY AND APPROXIMATIONS

Here we review some results presented in (Bullo and Leonard, 1997); a more detailed treatment is found in (Bullo *et al.*, 1997).

2.1 *Tools and models*

Let G denote a matrix Lie group with matrix multiplication as group operation. Let $\mathfrak{g}$ be its Lie algebra and let $\mathrm{ad}_\xi\, \eta = [\xi, \eta] = \xi\eta - \eta\xi$ denote the Lie bracket operation (i.e. matrix commutator). We assume that the set G is the Cartesian product of copies of the group of rigid motions SE(3) and its proper subgroups. This includes for example the rotation group SO(3) and its associated Lie algebra $\mathfrak{so}(3)$. On an arbitrary Lie group we can define a surjective map and local diffeomorphism called the *exponential map* $\exp : \mathfrak{g} \to G$. For example, given $x \in \mathbb{R}^3$, Rodrigues' formula gives

$$\exp_{\mathrm{SO}(3)}(\widehat{x}) = \mathrm{I} + \sin\|x\| \frac{\widehat{x}}{\|x\|} + (1 - \cos\|x\|) \frac{\widehat{x}^2}{\|x\|^2},$$

where the operator $\widehat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3)$ is defined via $\widehat{x} y \triangleq x \times y$ for all $x, y \in \mathbb{R}^3$. In an open neighborhood of the identity $\mathrm{I} \in G$, we define $x = \log(g) \in \mathfrak{g}$ to be the *exponential coordinates* of the group element g and we regard the logarithmic map as a local chart on the manifold G.

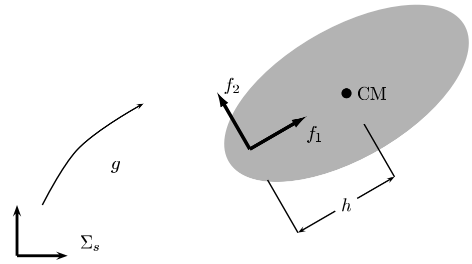

Fig. 1. Planar body with two forces applied at a point a distance h from the center of mass.

Definition 1. We describe a *mechanical control system on a Lie group* with the following objects: an n-dimensional Lie group G (defining the configuration space), an inertia tensor $\mathbb{I} : \mathfrak{g} \to \mathfrak{g}^*$ (defining the kinetic energy) and a set of input co-vectors $\{f_1, \ldots, f_m\} \subset \mathfrak{g}^*$ (defining the body-fixed forces). The system is said to be *underactuated* if the number of available input forces m is less than the number of degrees of freedom n.

Let $g \in G$ denote the configuration of the system and $\xi \in \mathfrak{g}$ the body-fixed velocity, so that the kinetic energy is $\frac{1}{2}\xi^T \mathbb{I} \xi$. The kinematic and dynamic equations of motion are

$$\dot{g} = g \cdot \xi \tag{1}$$

$$\mathbb{I}\dot{\xi} = \mathrm{ad}_\xi^* \, \mathbb{I}\xi + \sum_{i=1}^m f_i u_i(t), \tag{2}$$

where ad_ξ^* is the dual operator of ad_ξ and where the scalar input functions $\{u_i, i = 1, \ldots, m\}$ belong to the space of measurable functions $\mathcal{U}^m$. For convenience we define $b_i \triangleq \mathbb{I}^{-1} f_i$.

For any vector η with the property that $\mathrm{ad}_\eta^* \, \mathbb{I}\eta = 0$, the curve $t \in \mathbb{R} \mapsto (\exp(t\eta), \eta)$ is a solution to the system (1)–(2). These curves are studied in mechanics (Marsden and Ratiu, 1994) under the name of *relative equilibria* and describe motion that corresponds to constant body-fixed velocity for the uncontrolled system.

Next, we define the operation of *symmetric product* on the Lie algebra $\mathfrak{g}$. Given two vectors ξ, η on $\mathfrak{g}$, we define

$$\langle \xi : \eta \rangle \triangleq -\mathbb{I}^{-1}\big(\mathrm{ad}_\xi^* \, \mathbb{I}\eta + \mathrm{ad}_\eta^* \, \mathbb{I}\xi \big).$$

For example, on $\mathfrak{so}(3) \approx \mathbb{R}^3$ with the inertia tensor $\mathbb{J}$ and with the equality $\mathrm{ad}_\eta^* = -\hat{\xi}$, we compute $\langle \xi : \eta \rangle = \mathbb{J}^{-1}\big(\xi \times \mathbb{J}\eta + \eta \times \mathbb{J}\xi\big)$.

Example 2. (Planar body in a fluid). Let (θ, x, y) in SE(2) denote the configuration of the planar body and (ω, v_1, v_2) its body-fixed velocity. The kinetic energy is $KE = \frac{1}{2}J\omega^2 + \frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2$ where $\{J, m_1, m_2\}$ are positive constants. On $\mathfrak{se}(2)$ the adjoint operator is computed as

$$\mathrm{ad}_{(\omega, v_1, v_2)} = \begin{bmatrix} 0 & 0 & 0 \\ v_2 & 0 & -\omega \\ -v_1 & \omega & 0 \end{bmatrix}.$$

The two control inputs consist of forces applied at a distance h from the center of mass, see Figure 1. After inverting $\mathbb{I} = \mathrm{diag}\{J, m_1, m_2\}$, we have $b_1 = \frac{1}{m_1}\mathsf{e}_2$ and $b_2 = \frac{-h}{J}\mathsf{e}_1 + \frac{1}{m_2}\mathsf{e}_3$, where $\{\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3\}$ is the standard basis on $\mathbb{R}^3$. In coordinates the equations of motion (1)–(2) read

$$\begin{aligned} \dot\theta &= \omega & J\dot\omega &= (m_1 - m_2)v_1 v_2 - h u_2 \\ \dot x &= c_\theta v_1 - s_\theta v_2\,, & m_1 \dot v_1 &= m_2 \omega v_2 + u_1 \\ \dot y &= s_\theta v_1 + c_\theta v_2 & m_2 \dot v_2 &= -m_1 \omega v_1 + u_2. \end{aligned}$$

where $c_\theta = \cos(\theta)$ and $s_\theta = \sin(\theta)$.

2.2 *Local controllability*

For $T > 0$, a *solution* of the system (1)–(2), is a pair (g, u), where $g : [0, T] \to G$ is a piecewise smooth curve on G and u is an admissible input. Let $g_0 \in G$ and let $W \subset G \times \mathfrak{g}$ be a neighborhood of $(g_0, 0)$. For $T > 0$, set

$$\mathcal{R}^W(g_0, \leq T) = \bigcup_{0 \leq t \leq T} \{(g_1, \xi_1) \mid \exists \text{ a sol.}(g, \xi) \text{ s.t.}(g, \xi)(0) = (g_0, 0)$$
$$(g, \xi)(s) \in W \text{ for } s \in [0, t] \text{ and } (g, \xi)(t) = (g_1, \xi_1)\}.$$

Definition 3. The system (1)–(2) is *small-time locally controllable at g_0 and at zero velocity* if $\mathcal{R}^W(g_0, \leq T)$ contains a open set of $G \times \mathfrak{g}$ for all $T > 0$ and for all W, and if $(g_0, 0_{\mathfrak{g}})$ belongs to the interior of this set. If this holds for any $g_0 \in G$, the system is called *small-time locally controllable at zero velocity* (STLC at zero velocity).

Let $\mathcal{B} = \{b_1, \dots, b_m\}$ denote the family of input vectors. We define the *symmetric closure* of $\mathcal{B}$, denoted by $\overline{\mathrm{Sym}}(\mathcal{B})$, as the set of vectors obtained by taking iterated symmetric products of the vectors in $\mathcal{B}$. The *order* of an iterated product of factors from $\overline{\mathrm{Sym}}(\mathcal{B})$ is the total number of factors. We say that a symmetric product from $\overline{\mathrm{Sym}}(\mathcal{B})$ is *bad* if it contains an even number of each of the vectors in $\mathcal{B}$. Otherwise, we say that the symmetric product is *good*. For example, $\langle\langle b_1 : b_2\rangle : b_1\rangle$ has order three and is good, $\langle\langle\langle b_1 : b_2\rangle : b_2\rangle : b_1\rangle$ has order four and is bad.

Proposition 4. The system (1)–(2) is STLC at zero velocity if the subspace $\overline{\mathrm{Sym}}(\mathcal{B})$ has full rank and every bad symmetric product is a linear combination of lower-order good symmetric products.

Example 5. (Planar body in a fluid). Consider the planar body of Example 2 The only relevant non-vanishing symmetric products are

$$\langle b_1 : b_2\rangle = \frac{-h}{J m_2}\mathsf{e}_3 + \frac{m_2 - m_1}{J m_1 m_2}\mathsf{e}_1, \quad \langle b_2 : b_2\rangle = \frac{2h}{J m_1}\mathsf{e}_2,$$
$$\langle b_2 : \langle b_2 : b_2\rangle\rangle = \frac{-2h(m_1 - m_2)}{J^2 m_1 m_2}\mathsf{e}_1 + \frac{-2h^2}{J^2 m_2}\mathsf{e}_3.$$

The system is STLC at zero velocity, since the subspace generated by the vectors $\{b_1, b_2, \langle b_1 : b_2\rangle\}$ has full rank and the bad symmetric product $\langle b_2 : b_2\rangle$ is a linear combination of lower-order good products: $\langle b_2 : b_2\rangle = -(2h/J)b_1$.

Motivated by this example, we say that a system is *STLC at zero velocity with second-order symmetric products* if

(AC) The subspace $\mathrm{span}\{b_i, \langle b_j : b_k\rangle, j < k\}$ has full rank and each bad product $\langle b_i : b_i\rangle$ is a linear combination of the vectors $\{b_1, \dots, b_m\}$.

2.3 *Approximate solutions under small-amplitude forcing*

In this section we compute an approximate solution of system (1)–(2) under small-amplitude forcing. Given a vector-valued function $h(t)$, $t \in \mathbb{R}_+$, define its first integral function $\overline{h}(t)$, as the finite integral from 0 to t

$$\overline{h}(t) \triangleq \int_0^t h(\tau) d\tau.$$

Higher-order integrals, as for example $\overline{\overline{h}}(t)$, are defined recursively. In the following, we consider inputs of the form

$$u_i(t, \epsilon) = \epsilon u_i^1(t) + \epsilon^2 u_i^2(t)$$

where $0 < \epsilon << 1$ and where u_i^1, u_i^2 are $O(1)$. Accordingly we write the resultant forcing $\sum_i b_i u_i(t, \epsilon)$ as the sum of two terms of different order in ϵ

$$\begin{aligned} \sum_{i=1}^m b_i u_i(t, \epsilon) &= \sum_{i=1}^m b_i \left(\epsilon u_i^1(t) + \epsilon^2 u_i^2(t)\right) \\ &= \epsilon\, b^1(t) + \epsilon^2\, b^2(t), \end{aligned} \tag{3}$$

where we define $b^1(t) = \sum_{i=1}^m b_i u_i^1(t)$ and $b^2(t) = \sum_{i=1}^m b_i u_i^2(t)$. In the following, given any quantity $x(\epsilon)$, we let x^k denote the kth term in the Taylor expansion of $x(\epsilon)$ about $\epsilon = 0$; for example, we will write $\xi(t, \epsilon) = \epsilon \xi^1(t) + \epsilon^2 \xi^2(t) + O(\epsilon^3)$.

Proposition 6. Let $(g(t), \xi(t))$ be the solution to (1)–(2), let $\epsilon \ll 1$ and define the inputs as in (3). Let $x(t)$ be the exponential coordinates of $g(t)$, and let $\xi(0) = \epsilon \xi_0^1 + \epsilon^2 \xi_0^2$. Then for $t \in [0, 2\pi]$ it holds that $\xi(t, \epsilon) = \epsilon \xi^1(t) + \epsilon^2 \xi^2(t) + O(\epsilon^3)$, with

$$\xi^1(t) = \xi_0^1 + \overline{b^1}(t),$$
$$\xi^2(t) = \xi_0^2 - \langle \xi_0^1 : \xi_0^1\rangle \frac{t}{2} - \langle \xi_0^1 : \overline{\overline{b^1}}\rangle + \overline{\left(b^2 - \frac{\langle \overline{b^1} : \overline{b^1}\rangle}{2}\right)}$$

and $x(t, \epsilon) = \epsilon x^1(t) + \epsilon^2 x^2(t) + O(\epsilon^3)$, with

$$x^1(t) = \xi_0^1 t + \overline{\overline{b^1}}(t),$$
$$\begin{aligned} x^2(t) = \xi_0^2 t - \langle \xi_0^1 : \xi_0^1\rangle \frac{t^2}{4} + \overline{\overline{\left(b^2 - \tfrac{1}{2}\langle \overline{b^1} : \overline{b^1}\rangle\right)}}(t) \\ - \langle \xi_0^1 : \overline{\overline{\overline{b^1}}}(t)\rangle + \tfrac{1}{2}\overline{\left[\xi_0^1 + \overline{b^1}, \xi_0^1 t + \overline{\overline{b^1}}\right]}(t). \end{aligned}$$

Next we design inputs $\left(b^1(t), b^2(t)\right)$, that allow us to simplify the approximations above and steer the velocity of the system to an arbitrary value.

Lemma 7. (Inversion Algorithm). Let (AC) hold and let η be an arbitrary element in $\mathfrak{g}$.

(1) Set $N = m(m-1)/2$ and let $P = \{(j,k) \mid 1 \leq j < k \leq m\}$. Number the elements in P with $1, \dots, N$, and let $a(j,k)$ be the integer corresponding to (j,k). For $a = 1, \dots, N$, let

$$\psi_a(t) = \frac{1}{\sqrt{2\pi}}\Big(a\sin(at) - (a+N)\sin(a+N)t\Big).$$

(2) Because of (AC), the matrix with columns b_i and $\langle b_j : b_k \rangle$ has full rank. Via its pseudo-inverse, compute $(m+N)$ real numbers η_i and η_{jk} such that

$$\eta = \sum_{1\leq i \leq m} \eta_i b_i + \sum_{1\leq j<k\leq m} \eta_{jk} \langle b_j : b_k \rangle .$$

(3) Finally, set

$$b^1 = \sum_{j<k} \sqrt{|\eta_{jk}|}\Big(b_j - \operatorname{sign}(\eta_{jk}) b_k\Big)\psi_{a(j,k)}(t),$$

$$b^2 = \sum_i \frac{\eta_i}{2\pi} b_i + \sum_{j<k} \frac{|\eta_{jk}|}{4\pi}\Big(\langle b_j : b_j \rangle + \langle b_k : b_k \rangle\Big),$$

(4) and denote this procedure with

$$\left(b^1(t), b^2(t)\right) = \texttt{Inverse}(\eta) .$$

It can be shown that

$$\left(\overline{b^2 - \tfrac{1}{2}\langle \overline{b^1} : \overline{b^1} \rangle}\right)(2\pi) = \eta.$$

3. MOTION PRIMITIVES AND CONTROL ALGORITHMS

In this section we design motion control algorithms based on the approximations in Proposition 6 and the inversion algorithm in Lemma 7. We start by designing two primitive motion patterns, `Maintain-Vel` and `Change-Vel`.

3.1 *Primitives of motion*

We describe two basic maneuvers that each last 2π units of time. The parameter $\sigma \ll 1$ is a new small positive constant. To maintain a velocity of order $O(\sigma)$, an input of order $O(\sigma)$ suffices, while to obtain a change in velocity of order $O(\sigma)$, we employ a control input of order $O(\sqrt{\sigma})$. Each primitive is described in terms of initial configuration and velocity, input design, and final configuration and velocity.

`Maintain-Vel`$(\sigma, \xi_{\text{ref}})$: keeps the body velocity $\xi(t)$ close to a reference value $\sigma\xi_{\text{ref}}$. The initial state is

$$(g(0), \xi(0)) = (g_0, \sigma\xi_{\text{ref}} + \sigma^2\xi_{\text{err}}),$$

the input is designed as $\epsilon = \sigma$ and

$$(b^1, b^2) = \texttt{Inverse}(\pi \langle \xi_{\text{ref}} : \xi_{\text{ref}} \rangle - \xi_{\text{err}}) ,$$

and the final state is

$$\log(g_0^{-1} g(2\pi)) = 2\pi\sigma\xi_{\text{ref}} + \pi\sigma^2\xi_{\text{err}} + O(\sigma^3),$$
$$\xi(2\pi) = \sigma\xi_{\text{ref}} + O(\sigma^3). \qquad (4)$$

`Change-Vel`$(\sigma, \xi_{\text{final}})$: steer the body velocity $\xi(t)$ to a final value $\sigma\xi_{\text{final}}$. The initial state is

$$(g(0), \xi(0)) = (g_0, \sigma\xi_0),$$

the input is designed as $\epsilon = \sqrt{\sigma}$ and

$$(b^1, b^2) = \texttt{Inverse}(\xi_{\text{final}} - \xi_0) ,$$

and the final state is

$$\log(g_0^{-1} g(2\pi)) = \pi\sigma(\xi_0 + \xi_{\text{final}}) + O(\sigma^{3/2})$$
$$\xi(2\pi) = \sigma\xi_{\text{final}} + O(\sigma^2). \qquad (5)$$

The statements (4) and (5) are proved via an application of the results in Proposition 6 and in Lemma 7. Next we compute estimates of final configurations after multiple intervals. The following result is a consequence of the Campbell–Baker–Hausdorff formula (Marsden and Ratiu, 1994).

Lemma 8. Let $\sigma \ll 1$ be a positive constant, let $g_0, g_1 \in G$, and set $y_0 = \log(g_0) \in \mathfrak{g}$, $y_1 = \log(g_1) \in \mathfrak{g}$. If the vector $[y_0, y_1]$ is higher order in σ than $(y_0 + y_1)$, then

$$\log(g_0 g_1) = y_0 + y_1 + O([y_0, y_1]).$$

Finally we evaluate costs associated with the two primitives. The magnitude of control input is

$$\|\pi \langle \xi_{\text{ref}} : \xi_{\text{ref}} \rangle - \xi_{\text{err}}\| O(\sigma),$$

during a `Maintain-Vel`$(\sigma, \xi_{\text{ref}})$ primitive and

$$\|\xi_{\text{final}} - \xi_0\| O(\sqrt{\sigma}),$$

during a `Change-Vel`$(\sigma, \xi_{\text{final}})$ primitive.

3.2 *Control algorithms*

We present three algorithms to solve various motion control problems. These algorithms combine the two motion primitives with a discrete-time feedback. This makes the approximations hold over multiple time intervals, as, for example, over a time interval of order $1/\sigma$.

Point-to-point reconfiguration problem This motion task reconfigures the system, i.e. changes its position and orientation, starting and ending at zero velocity. We assume that the initial state is $(g(0), \xi(0)) = (g_0, 0_{\mathfrak{g}})$ and the final desired state is $(g_1, 0_{\mathfrak{g}})$. For simplicity, we require $\log(g_0^{-1} g_1)$ to be well defined, even though this assumption can be removed. On SO(3) the logarithm is well defined whenever the change in attitude is less that π.

Table 1. Constant Velocity Algorithm

Goal: steer from $(g_0, 0_{\mathfrak{g}})$ to $(g_1, 0_{\mathfrak{g}})$.
Arguments: (g_0, g_1, σ).

$N \Leftarrow \text{Floor}(\|\log(g_0^{-1} g_1)\|/(2\pi\sigma))$
  {Floor(x) is the greatest integer $\leq x$.}
$\xi_{\text{nom}} \Leftarrow \log(g_0^{-1} g_1)/(2\pi\sigma N)$
`Change-Vel`$(\sigma, \xi_{\text{nom}})$
for $k = 1$ to $(N-1)$ **do**
  `Maintain-Vel`$(\sigma, \xi_{\text{nom}})$
end for
`Change-Vel`$(\sigma, 0_{\mathfrak{g}})$

The algorithm consists of three steps. Over the first time interval, we change the velocity to an appropriate reference value. We then maintain the velocity close to this constant reference value for an appropriate number of periods. Finally, we stop the system when close to the desired configuration. The details are described in Table 1.

Lemma 9. (Constant Velocity Algorithm). Let σ be a sufficiently small positive constant and let $(g(0), \xi(0)) = (g_0, O(\sigma^2))$ and let g_1 be a group element such that $\log(g_0^{-1} g_1)$ is well defined. Let $N \in \mathbb{N}$ and the inputs $(b^1, b^2)(t)$ for $t \in [0, 2(N+1)\pi]$ be determined according to the algorithm in Table 1. At final time it holds

$$\log\Big(g(2(N+1)\pi)^{-1} g_1\Big) = O(\sigma^{3/2}),$$
$$\xi\Big(2(N+1)\pi\Big) = O(\sigma^2).$$

The final state is not exactly as desired, instead there are errors of order $O(\sigma^{3/2})$ and $O(\sigma^2)$. This undesirable feature can be compensated for by solving the point stabilization problem.

Point stabilization problem This motion task asymptotically stabilizes the configuration $g(t)$ to a desired value that we assume without loss of generality to be the identity. Convergence is ensured as long as $\|(\log(g(0)), \xi(0))\|$ is small enough.

The key idea of the algorithm is to iterate the following procedure: measure the state at time t_k and design control inputs that try to steer the state to the desired value $(\mathrm{I}, 0_{\mathfrak{g}})$ at time $t_{k+1} = t_k + 4\pi$. Since we impose two requirements, one on the final configuration and one on the final velocity, two calls to the `Change-Vel` primitive are needed. The details are described in Table 2.

Lemma 10. (Exponential Stabilization Algorithm). Let $\|(\log(g(0)), \xi(0))\| < \sigma$ be sufficiently small. Let the inputs $(b^1(t), b^2(t))$ be determined according to the algorithm in Table 2 and let $t_k = 4k\pi$. Then there exists a $\lambda > 0$ such that for all $k \in \mathbb{N}$

$$\big\|\big(\log(g(t_k)), \xi(t_k)\big)\big\| \leq \big\|\big(\log(g(0)), \xi(0)\big)\big\| e^{-\lambda k}.$$

Table 2. Exponential Stabilization Alg.

Goal: steer the state to $(\mathrm{I}, 0_{\mathfrak{g}})$ as $t \to \infty$.

for $k = 1$ to $+\infty$ **do**
  $t_k \Leftarrow 4k\pi$ {t_k is the current time}
  $\sigma_k \Leftarrow \|(\log(g(t_k)), \xi(t_k)))\|$
  $\xi_{\text{tmp}} = -\big(\log(g(t_k)) + \pi\xi(t_k)\big)/(2\pi\sigma_k)$
  `Change-Vel`$(\sigma_k, \xi_{\text{tmp}})$
  `Change-Vel`$(\sigma_k, 0_{\mathfrak{g}})$
end for

Table 3. Static Interpolation Algorithm

Goal: steer through points $\{g_i\}$.
Arguments: $(g_0, g_1, \ldots, g_M, \sigma)$.

for $j = 1$ to M **do**
  $g_{\text{tmp},j} \Leftarrow g(t)\exp(\pi\xi(t))$
  $N_j \Leftarrow \text{Floor}(\|\log(g_{\text{tmp},j}^{-1} g_j)\|/(2\pi\sigma))$
  $\xi_{\text{nom},j} \Leftarrow \log(g_{\text{tmp},j}^{-1} g_j)/(2\pi\sigma N_j)$
  `Change-Vel`$(\sigma, \xi_{\text{nom},j})$
  for $k = 1$ to $(N_j - 1)$ **do**
    `Maintain-Vel`$(\sigma, \xi_{\text{nom},j})$
  end for
end for
`Change-Vel`$(\sigma, 0_{\mathfrak{g}})$

Additionally, for $t \in [4k\pi, 4(k+1)\pi]$ it holds $\big\|\big(\log(g(t), \xi(t)\big)\big\| = O(e^{-\lambda k/2})$.

Static Interpolation problem This motion task steers the system's configuration along a path connecting the set of the ordered points $\{g_0, \ldots, g_M\}$. As above, we require $\log(g_{k-1}^{-1} g_k)$ to be well defined for $1 \leq k \leq M$. Roughly speaking, the algorithm consists of M repeated constant velocity maneuvers but without stopping in between points. The details are described in Table 3. It can be shown that the configuration $g(t)$ follows a path passing through the points $\{g_0, g_1, \ldots, g_M\}$ with an error of order σ.

3.2.1. *Interpolating sequences of relative equilibria versus constant velocity motions* The Constant Velocity and the Static Interpolation Algorithms provide two different solutions to the reconfiguration problem. These two algorithms can be compared on the basis of an input cost of the form

$$\|u\|_{[0,T]} = \int_0^T L(u(t))dt,$$

where $T = T(\sigma)$ is the time required to complete the maneuver and $L : \mathcal{U}^m \mapsto \mathbb{R}$ is a cost on the space of input functions. We let g_i and g_f denote initial and final (desired) configurations and we let $\mathcal{P} = \{g_0 = g_i, g_1, \ldots, g_M = g_f\}$ be a sequence of configurations such that $\log(g_{j-1}^{-1} g_j)$ is a relative equilibrium vector for all $j = 1, \ldots, M$. Recall that $\eta \in \mathfrak{g}$ is a relative equilibrium vector if $\langle \eta : \eta \rangle$ vanishes.

(1) The Constant Velocity Algorithm to go from g_0 to g_f involves 2 calls to the `Change-Vel`

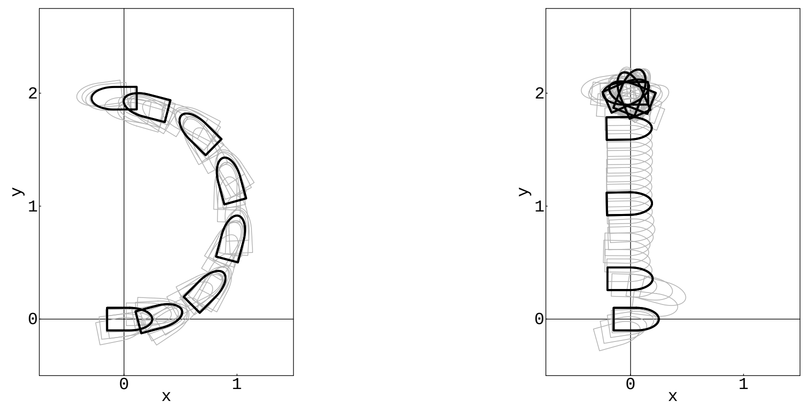

Fig. 2. Constant Velocity (left) and Static Interpolation (picture) algorithms. The bullet-shaped objects represent the planar body's location: darker objects correspond to beginning and end of a primitive.

primitive and (N–1) calls to the `Maintain-Vel` primitive. The cost of the complete maneuver can be computed as

$$\|u\|_{[0,T]} = O(\sqrt{\sigma}) + (N-1)\|\langle\xi_{\text{nom}} : \xi_{\text{nom}}\rangle\|O(\sigma) = O(1),$$

since $\|\langle\xi_{\text{nom}} : \xi_{\text{nom}}\rangle\| = O(1)$ and $N = O(1/\sigma)$.

(2) The Interpolation Algorithm applied to the set of configurations $\mathcal{P}$ involves $(M+2)$ calls to the `Change-Vel` primitive and $(\sum_{j=1}^{M} N_j)$ calls to the `Maintain-Vel` primitive. With the notation in Table 3

$$\|u\|_{[0,T]} = (M+2)O(\sqrt{\sigma}) + (\textstyle\sum_j N_j)\|\langle\xi_{\text{nom},j} : \xi_{\text{nom},j}\rangle\|O(\sigma).$$

Since the configuration $g(t)$ follows the path determined by set $\mathcal{P}$ with an error of order σ, and since $\log(g_{j-1}^{-1} g_j)$ is a relative equilibrium vector, it can be shown that $\langle\xi_{\text{nom},j} : \xi_{\text{nom},j}\rangle = O(\sigma)$. Summarizing

$$\|u\|_{[0,T]} = (M+2)O(\sqrt{\sigma}) + (\textstyle\sum_j N_j)O(\sigma^2) = O(\sqrt{\sigma}).$$

We conclude that for small σ (or equivalently, for long final times $T = O(1/\sigma)$), moving along a set of relative equilibria is a more efficient strategy than the Constant Velocity Algorithm.

3.3 *Numerical simulations*

The algorithms have been implemented on the planar body in Example 2. The parameter values in normalized units were chosen to be $J=1, m_1 = .6, m_2 = 1, h = 2$. For both the Constant Velocity Algorithm and the Static Interpolation Algorithm, we let the initial configuration be the identity and the final (desired) configuration consist of a rotation of π and a translation of 2 units along the y-axis. Equivalently, we set $g_{\text{initial}} = (0,0,0)$ and $g_{\text{final}} = (\pi, 0, 2)$. We let the parameter σ vary in the $[.001, .5]$ range, and we present here results for the $\sigma = .1$ case. For all three algorithms, the numerical results were in agreement with the theoretical analysis presented above.

The left picture in Figure 2 corresponds to the Constant Velocity Algorithm and illustrates how the configuration variables evolve along a screw motion toward the desired configuration. The right picture in Figure 2 corresponds to the Static Interpolation Algoritm. The initial and final (desired) configurations are the same as in the previous run. The set of ordered configuration points is $\{(0,0,0), (0,0,2), (\pi,0,2)\}$. The path in the x, y plane consists of a straight line and a rotation.

4. REFERENCES

Bullo, F. and N. E. Leonard (1997). Motion control for underactuated mechanical systems on Lie groups. In: *Proc ECC*. Brussels, Belgium. p. 480.

Bullo, F., N. E. Leonard and A. D. Lewis (1997). Controllability and motion algorithms for underactuated lagrangian systems on Lie groups. Technical Report CDS 97–013. California Institute of Technology. Available at `http://www.cds.caltech.edu`.

Lafferriere, G. and H. J. Sussmann (1991). Motion planning for controllable systems without drift. In: *Proc ICRA*. Sacramento, CA. pp. 1148–1153.

Leonard, N. E. and P. S. Krishnaprasad (1995). Motion control of drift-free, left-invariant systems on Lie groups. *IEEE TAC* **40**(9), 1539–1554.

Lewis, A. D. and R. M. Murray (1997). Configuration controllability of simple mechanical control systems. *SIAM J. Control and Optimization* **35**(3), 766–790.

Marsden, J. E. and T. S. Ratiu (1994). *Introduction to Mechanics and Symmetry*. Springer Verlag. New York, NY.

Sussmann, H. J. (1987). A general theorem on local controllability. *SIAM J. Control and Optimization* **25**(1), 158–194.