University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS) | Department of Computer & Information Science

January 1999

Steering for a Class of Dynamic Nonholonomic Systems

James P. Ostrowski
University of Pennsylvania, jpo@grasp.cis.upenn.edu

Follow this and additional works at: https://repository.upenn.edu/cis_reports

Recommended Citation

James P. Ostrowski, "Steering for a Class of Dynamic Nonholonomic Systems", . January 1999.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-99-22.

This paper is posted at ScholarlyCommons. https://repository.upenn.edu/cis_reports/161
For more information, please contact repository@pobox.upenn.edu.

Steering for a Class of Dynamic Nonholonomic Systems

Abstract

In this paper we derive control algorithms for a class of dynamic nonholonomic steering problems, characterized as mechanical systems with nonholonomic constraints and symmetries. Recent research in geometric mechanics has led to a single, simplified framework that describes this class of systems, which includes examples such as wheeled mobile robots; undulatory robotic and biological locomotion systems, such as paramecia, inchworms, and snakes; and the reorientation of satellites and underwater vehicles. This geometric framework has also been applied to more unusual examples, such as the snakeboard robot, bicycles, the wobblestone, and the reorientation of a falling cat. We use this geometric framework as a basis for developing two types of control algorithms for such systems. The first is geared towards local controllability, using a perturbation approach to establish results similar to steering using sinusoids. The second method utilizes these results in applying more traditional steering algorithms for mobile robots, and is directed towards generating more non-local control methods of steering for this class of systems.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-99-22.

This technical report is available at ScholarlyCommons: https://repository.upenn.edu/cis_reports/161

Steering for a class of dynamic nonholonomic systems

Jim Ostrowski
University of Pennsylvania
Philadelphia, PA 19104-6315
jpo@grip.cis.upenn.edu

Abstract

In this paper we derive control algorithms for a class of dynamic nonholonomic steering problems, characterized as mechanical systems with nonholonomic constraints and symmetries. Recent research in geometric mechanics has led to a single, simplified framework that describes this class of systems, which includes examples such as wheeled mobile robots; undulatory robotic and biological locomotion systems, such as paramecia, inchworms, and snakes; and the reorientation of satellites and underwater vehicles. This geometric framework has also been applied to more unusual examples, such as the snakeboard robot, bicycles, the wobblestone, and the reorientation of a falling cat. We use this geometric framework as a basis for developing two types of control algorithms for such systems. The first is geared towards local controllability, using a perturbation approach to establish results similar to steering using sinusoids. The second method utilizes these results in applying more traditional steering algorithms for mobile robots, and is directed towards generating more non-local control methods of steering for this class of systems.

1 Introduction

Mechanical systems have traditionally provided a fertile area of study for researchers interested in nonlinear control, due to the inherent nonlinearities and the Lagrangian structure of these systems. Recently, a great deal of emphasis has been placed on studying systems with nonholonomic (non-integrable) constraints, including mobile wheeled robots and multiple-trailer vehicles, where the wheels provide a no-slip velocity constraint. For the purposes of controls, however, these systems are very often treated as kinematic systems, i.e., the dynamics of these mechanical systems are assumed to be inverted out. Very often this assumption is quite valid, but frequently it is not [13]. There are also a growing number of systems in which this type of assumption is not even approximately valid. We focus in this paper on one such class of systems; namely, the class of underactuated systems with nonholonomic constraints and symmetries.

Making use of modern advances in geometric mechanics, researchers have made great progress in analyzing the mechanics of locomotion. This problem asks the fundamental question of how does a system use its control inputs to effect motion from one place to another. By utilizing the inherent mathematical structure found in these types of problems, one can formulate the dynamics of a wide variety of locomotion problems in a very intuitively appealing and insightful manner. Doing so leads to a stronger comprehension of the *mechanics* of locomotion, which in turn provides a foundation for studying issues of *control* for such systems. It is clear [4, 8, 12, 20] that the geometric tools used to formulate the mechanics can also provide a basis for unlocking the answers to many of the questions regarding the control theoretic issues involved. A general introductory review to some of the ideas of geometric mechanics in motion control is given in [15].

An important by-product of the mechanics research in locomotion has been the development of a theoretical bridge between systems with two different types of nonholonomic constraints. On one hand, there are systems with *external* (often called *kinematic*) constraints which include wheeled vehicles [17], grasping with point-finger contacts, and some models of snakes [11], paramecia [8], and even legged locomotion [8]. Recent work by Kelly and Murray [8] provides kinematically constrained models for a wide range of systems that locomote, and controllability results for these types of systems. However, the models are again restricted

to purely *kinematic* systems, which require active input controls to generate movement. A kinematically constrained body in motion will remain in motion only if its control inputs are continually active. Thus it is not possible to build momentum or to "coast"— it is exactly this component of locomotion that has been added in the models considered here.

A characteristic of all of these systems, however, is that they possess invariances with respect to Lie group symmetries [8, 12, 16, 20]. For locomotive systems, this is often a "pick-and-place" symmetry, whereby the rigid body dynamics are invariant with respect to inertial positioning. In the absence of external constraints, these invariances imply the existence of what can be thought of as "internal," or *dynamic*, constraints on the system, which most often take the form of momentum conservation laws. Examples of systems with momentum-type nonholonomic constraints include satellites in space [10, 27] and the "falling cat" [16].

Naturally, there exist problems for which the kinematics and dynamics interact in a nontrivial manner. Examples of these problems fall generally into two realms: one in which certain of the conservation laws may remain after the addition of constraints, such as in the rolling penny or the constrained particle; and one in which the conservation laws are transformed into what Bloch et al. term a *generalized momentum equation* [2], where the momenta are governed by a differential equation. There is strong evidence to suggest that many different modes of locomotion (such as undulatory, legged, etc.) are governed by equations of this form. What is present in the case of mixed constraints (i.e., kinematic *and* dynamic) is the ability to change the momentum of a locomotive body. This is crucial for many types of locomotion, such as running or swimming. Thus, by using internal, *shape* controls it is possible not only to change the position of the system, but to generate velocities and hence truly *locomote* in a dynamic sense. The current study focuses on introducing methods for controlling and planning trajectories (steering) for such dynamic systems. As an example, we will examine the snakeboard model, which has been an important motivating example behind the theoretical progress for this mixed kinematic and dynamic constraint case [2, 22].

In this paper, we use perturbation methods to derive new results on constructing open-loop controls for dynamic system using cyclic inputs. By making the assumption that the amplitudes of the inputs are small, we can introduce a scaling parameter into the system. The use of perturbation techniques has proven to be useful in analyzing several types of kinematic and dynamic control systems [4, 12, 24].

We introduce two methods for generating motions for systems involving both kinematic and dynamic components. In Section 2, we briefly review a few of the necessary concepts, and then highlight some of the more recent theoretical results on controllability and constructive control algorithms. In Sections 3, 4, and 5, we develop two new methods for controlling dynamic nonholonomic systems. The first is more general, and utilizes a perturbation approach (developed in Section 3) to isolate important integral relations between the input controls and the resultant motion. The tools for building control laws presented in Section 4 are primarily for local control of a system around an equilibrium point. We also present simulations using the snakeboard example. In Section 5, we present a heuristic algorithm for steering of a restricted class of dynamic nonholonomic systems, characterized by car-like steering problems. This generalizes the notion of steering to systems for which the velocity (or momentum) of the system is not directly controllable. Finally, in Section 6, we discuss some of the central issues in this research, and mention some potential avenues for future investigation.

2 Background and problem formulation

This paper targets the generation of constructive open-loop controls for a class of dynamic steering problems. As such, we will not focus on the underlying mechanics of the problem. Instead, we provide a brief overview and motivation of the geometric structure of nonholonomic mechanical systems with symmetries, and refer the reader to the various works in the literature [2, 8, 11, 19, 21].

The study of most steering problems to date implicitly involves the control of motion along Lie groups. The use of Lie groups in the context of mechanics also leads to the study of symmetries. The principal motivation for using Lie groups arises from our studies of robotic locomotion, where displacements occur in some subgroup of $SE(3)$, most often translation and rotation in the plane, $SE(2)$, or rigid body rotation, $SO(3)$. However, the analysis here is valid for general mechanical systems which have some or all of their dynamics evolving on a Lie group.

Formally, the *position* of a robot's body-fixed frame is considered as a *left translation* within a Lie group,

G. The remaining components of the system are assumed to be controllable,[1] and these configuration variables will be represented by a manifold M. Most often for locomotion systems, these variables will describe the internal *shape* of the system. Thus, the configuration manifold will be the product manifold given by $Q = G \times M$ and the left translation induces a *left action*, Φ, of G on Q. For those familiar with the mechanics literature, the manifold Q is said to define define a *trivial principal fiber bundle* with *fibers*, G, over a *base* space, M. We will assume the coordinates on Q to be decomposable into fiber and base coordinates, i.e., for $q \in Q$, we can write $q = (g, r) \in G \times M$. The action, $\Phi_h : Q \to Q$ of an element h on $q = (g, r) \in Q$ is then given by $\Phi_h(q) = (hg, r)$, where hg represents the left action of G on itself.

Since we are working with mechanical systems, we will assume the existence of a Lagrangian function, $L(q, \dot{q})$, on TQ. Additionally, we can model external constraints, such as might occur due to no-slip wheel conditions (for wheeled mobile robots) and viscous friction (for sliding motions). Let us restrict our attention to constraints that are linear in the velocities. We write these as a vector-valued set of k equations:

$$\omega_j^i(q)\dot{q}^j = 0, \qquad \text{for} \quad i = 1 \ldots k. \tag{2.1}$$

This class of constraints includes most commonly investigated nonholonomic constraints.

In most of the locomotion and steering problems studied to date the Lagrangian function and the constraints are invariant with respect to the Lie group. In other words, they are independent of translation and rotation of the Lie group. For the Lagrangian function, this implies that $L(\Phi_h q, D_q \Phi_h \dot{q}) = L(q, \dot{q})$ (for a more complete definition, see [2, 21]). Physically this implies that the dynamics of the system do not depend on the initial placement of the body frame. Requirements for invariance of the constraints follow similarly. A very important consequence of the invariance of the system with respect to the group action, Φ_h, is that we can study all of the dynamics on the tangent space of G simply by analyzing the system at the tangent space to G at the group identity, T_eG. In other words, we pull all velocity vectors $\dot{g}$ back to the identity, as $\xi = g^{-1}\dot{g}$. For those familiar with robotics, this simply represents a twist in $SE(n)$, or a velocity viewed in a body-fixed frame. This tangent space is important because of its special structure as a Lie algebra. We denote by $\mathfrak{g}$ the Lie algebra of a Lie group, G. All Lie algebras come endowed with a bracket operation, called the *Lie bracket*, which we will denote by $[\cdot, \cdot]_\mathfrak{g} : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$. The Lie bracket on the Lie algebra is directly related to the Lie bracket traditionally used in nonlinear control theory to study commutativity of vector fields: $[\xi, \eta]_\mathfrak{g} = g^{-1}[g\xi, g\eta]$, for $\xi, \eta \in \mathfrak{g}$, $[\cdot, \cdot]$ the Lie bracket of vector fields on G, and using a slight, but common, abuse of notation of $g\xi$ to denote the lifted action of $g \in G$ on $\xi \in \mathfrak{g} = T_eG$.

Although Lagrange multipliers can be used to eliminate the constraints and develop the dynamics for such systems, a geometric interpretation has been found to be quite useful. In the presence of symmetries, the equations of motion can be transformed into the following form [2, 19]:

$$g^{-1}\dot{g} = -\mathbb{A}(r)\dot{r} + I^{-1}(r)p, \tag{2.2}$$

$$\dot{p} = \dot{r}^T\alpha(r)\dot{r} + p^T\beta(r)\dot{r} + p^T\gamma(r)p, \tag{2.3}$$

$$\dot{r} = u. \tag{2.4}$$

These equations, of course, deserve a good deal of comment; however, we will only examine them briefly here, since the focus is on control (to gain a much better insight into these equations, the reader is referred to the paper by Bloch et al. [2], and examples from robotics explored in [11, 21]). Eqs. 2.2 and 2.4 are the *fiber* and *base* equations, respectively. They define velocity vectors for the configuration variables. Eq. 2.3 is called the *generalized momentum equation*, where p is a momentum vector associated with the momentum along each of the kinematically unconstrained fiber directions. Notice that in Eq. 2.4 we have assumed the base (shape) space to be fully controllable, with velocity level inputs, u (see [13, 19] for more details).

Notice that by using the symmetries and writing the equations in a body-fixed frame we can pull the group variable g out of the equation— this greatly helps to simplify the necessary calculations. Notice also the central role played by the term $\mathbb{A}(r)$ in Eq. 2.2. In the language of geometric mechanics, $\mathbb{A}$ is said to define a (local) *connection* on Q. In investigating issues of controllability, the connection is extremely useful, because it defines the role of the control inputs (as changes in shape) in generating spatial motion along the fiber (i.e., changes in position).

[1] A very interesting class of systems, which are not considered here, are those for which the shape space is not fully controlled, such as for the wobblestone [15].

The term $I^{-1}p$ determines the effect of the momentum on the fiber equations. I is called the *locked inertia tensor*. Its development is beyond the scope of this paper, but essentially I corresponds to the inertia of the system given a particular (locked) configuration of the shape variables. For the terms, α, β, and γ of the generalized momentum equation, we mention only that they are strictly functions of the base variables, r, and so the generalized momentum equation can be written as a function *only* of the base and momentum variables [21]. Notice that Eqs. 2.2–2.4 are not in standard form for an affine control system, since the input u enters in quadratically in Eq. 2.3 (though a dynamic extension of $\ddot{r} = u$ can be used, as in [20]). The analysis presented here, however, does not require this.

Existing controllability results and control algorithms Let $\mathcal{R}^V(q_0, T)$ denote the set of reachable points in Q from q_0 at time $T > 0$, using admissible controls, $u(t)$, and such that the trajectories remain in the neighborhood V of q_0 for all $t \leq T$. Furthermore, let

$$\mathcal{R}^V_T(q_0) = \cup_{t \leq T} \mathcal{R}^V(q_0, t)$$

be the set of all reachable points from q_0 within time T. From the definitions of reachable sets, we are naturally led to the definitions of *local accessibility* [18], where for all $q \in Q$, $R^V_T(q)$ contains a non-empty open set of Q for all neighborhoods V of q and all $T > 0$; and *small-time local controllability* (STLC) [26], when for any neighborhood V, time $T > 0$ and $q \in Q$, q is an interior point of $\mathcal{R}^V_T(q)$ for all $T > 0$.

For driftless systems, local accessibility and local controllability are equivalent. Notice, however, that the general types of systems in which we are interested will *require* the presence of a drift vector field, since this is how the momenta enter into the dynamic equations (notice, for example, the $I^{-1}p$ term in Eq. 2.2). We will discuss in Section 4 control algorithms for systems that are STLC, and in Section 5 a method for steering systems that need only be locally accessible, based on the notion of non-local "controllability" for mobile robots.

Motion control on Lie groups As mentioned above, an interesting class of problems are kinematic systems that evolve on a principal fiber bundle– these are generally referred to as *principal kinematic* systems. Assuming that the shape variables are directly controlled, the motion on the group is kinematically related to these motions via Eq. 2.2:

$$\begin{aligned} g^{-1}\dot{g} &= -\mathbb{A}(r)\dot{r} \\ \dot{r} &= u. \end{aligned} \tag{2.5}$$

By using the special structure provided by the Lie group symmetries, Kelly and Murray [8] were able to derive straightforward computational conditions for controllability and suggest methods for generating desired trajectories. They show that the controllability of a kinematic system can be determined solely from the local form of the connection, $\mathbb{A}$, its curvature, and higher covariant derivatives. The reader unfamiliar with exterior derivatives of differential forms is referred to [1].

DEFINITION 2.1 Given a local connection form, $\mathbb{A}$, on M, the *local curvature form* is the 2-form $D\mathbb{A}$ determined by evaluating the exterior derivative of $\mathbb{A}$ on horizontal vectors. In our setting, this implies

$$D\mathbb{A}(X, Y) = d\mathbb{A}(X, Y) - [\mathbb{A}(X), \mathbb{A}(Y)]_{\mathfrak{g}}, \tag{2.6}$$

where $X, Y \in \mathfrak{X}(M)$ are base vector fields, where

$$(d\mathbb{A})_{ij} = \frac{\partial \mathbb{A}_j}{\partial r^i} - \frac{\partial \mathbb{A}_i}{\partial r^j}.$$

One can then construct a series of subspaces of $\mathfrak{g}$ given by repeatedly taking higher derivatives of the connection:

$$\begin{aligned} \mathfrak{h}_1 &= \text{span}\{\mathbb{A}(X) : X \in T_rM\}, \\ \mathfrak{h}_2 &= \text{span}\{D\mathbb{A}(X, Y) : X, Y \in T_rM\}, \\ \mathfrak{h}_3 &= \text{span}\{L_Z D\mathbb{A}(X, Y) - [\mathbb{A}(Z), D\mathbb{A}(X, Y)]_{\mathfrak{g}} : X, Y, Z \in T_rM\}, \ldots \\ \mathfrak{h}_k &= \text{span}\{L_X \xi - [\mathbb{A}(Z), \xi]_{\mathfrak{g}}, [\eta, \xi]_{\mathfrak{g}} : X \in T_rM, \xi \in \mathfrak{h}_{k-1}, \eta \in \mathfrak{h}_2 \oplus \cdots \oplus \mathfrak{h}_{k-1}\}, \end{aligned} \tag{2.7}$$

where L denotes the Lie derivative of a Lie algebra-valued function on the shape space, M, with respect to a vector field on M.

Notice that in the above equations, the connection has been placed in the appropriate mathematical context as a Lie algebra-valued one-form on M. Thus, derivatives of $\mathbb{A}$ will take their values in $\mathfrak{g}$ when evaluated along the appropriate vector fields on M.

PROPOSITION 2.2 *[8] The system given by Eqs. 2.5 is locally controllable at $q \in Q$ if and only if*

$$\mathfrak{g} = \mathfrak{h}_2 \oplus \mathfrak{h}_3 \oplus \cdots .$$

Similar to the case restricted to a Lie group, the subspaces $\mathfrak{h}_k \subset \mathfrak{g}$ play an important role in generating open loop controls for kinematic systems. Providing an extension to work done by Leonard and Krishnaprasad [12], Radford and Burdick [24] have developed area rules for generating motion related to exterior derivatives of $\mathbb{A}$. They showed that for systems of this form, it is possible to relate the motion generated by each order of the Lie bracket to an integral on the input space. To be more concrete, it was shown that a motion in the direction given by $[\mathbb{A}_i, \mathbb{A}_j]_\mathfrak{g}$ can be generated using cyclic inputs u^i and u^j, and that the resultant motion is proportional to the area enclosed in the shape space by the inputs u^i and u^j (actually, the area enclosed by the integral of the inputs $\tilde{u}^i(t) = \int_0^t u^i d\tau$). That is, the magnitude of the motion is proportional to Area $= \int (u^i \tilde{u}^j - u^j \tilde{u}^i) d\tau$. Higher level iterated Lie brackets can also be controlled using similar geometric principles; for example, moment integrals of $\tilde{u}^i$, $\tilde{u}^j$, and $\tilde{u}^k$ give rise to motion proportional to brackets of the form $[\mathbb{A}_i, [\mathbb{A}_j, \mathbb{A}_k]_\mathfrak{g}]_\mathfrak{g}$.

Local accessibility and controllability tests for the mixed case As discussed above, the results of Bloch et al. [2] and the accompanying work by Ostrowski [19] show that the equations found in the principal kinematic case extend quite naturally to the situation in which the constraints do not wipe out all the dynamics. The formulation of such dynamic nonholonomic systems involves the introduction of a nonholonomic momentum equation (c.f., Eq. 2.3). Using the controllability tests for principal kinematic systems as a base, we were able to derive accessibility and controllability tests for dynamic nonholonomic systems [20]. In this case, we use $\mathbb{A}$ to denote the nonholonomic connection defined in Eq. 2.2.

PROPOSITION 2.3 *If α is onto (as a map to the momentum tangent space), and*

$$\mathfrak{g} = \mathfrak{h}_2 + \mathfrak{h}_3 + \cdots ,$$

then the system given by Eqs. 2.2–2.4 is locally accessible.

For nonlinear systems with drift, however, local accessibility may be quite different from local controllability. In order to show local controllability (STLC), we use conditions given by Sussman [26] (which are the strongest we are currently aware of). For further details on this construction, please refer to [3, 20, 26].

PROPOSITION 2.4 *If the conditions of Proposition 2.3 are satisfied and each $\alpha_{ii} \equiv 0$ for $i = 1, \ldots, m$ (no summation over i), then the system given by Eqs. 2.2–2.4 is small-time locally controllable (STLC) from all equilibrium points in Q.*

3 A perturbation approach to constructive controls

Motivated by recent work by Leonard, Krishnaprasad, and Bullo [4, 12], a perturbation approach has been used to analyze the response of such dynamic nonholonomic systems to small amplitude cyclic inputs. It appears that these techniques provide a powerful mechanism for building open loop controls for nonlinear control systems. The reader is referred to Khalil [9] for more details on the use of perturbation methods.

We begin with the general form of the equations for a dynamic nonholonomic system, but with the restriction that the terms in the momentum equation that are quadratic in momenta be zero (that is, $\gamma = 0$

from Eq. 2.3). We re-write these equations here using indicial notation, since this will be important for making sense of the results given below:

$$\xi^a = (g^{-1}\dot{g})^a = -\mathbb{A}_i^a(r)\dot{r}^i + (I^{-1})^{ab}(r)p_b, \tag{3.8}$$

$$\dot{p}_a = \alpha_{aij}(r)\dot{r}^i\dot{r}^j + \beta_{ai}^b(r)\dot{r}^i p_b. \tag{3.9}$$

This represents the cases in which we are interested, namely dynamic steering systems like the bicycle, snakeboard, roller racer, etc. In fact, it can be shown that in all cases where the nonholonomic momentum is one-dimensional, the term γ will be identically zero [2]. The case in which γ is the only nonzero term in the momentum equation is treated implicitly in [4].

Next, consider inputs of the form:

$$r(t) = r_0 + \epsilon u(t). \tag{3.10}$$

These are slightly different than those defined by Leonard and Krishnaprasad as $\dot{r} = \epsilon u(t)$— we will see that our choice of r in Eq. 3.10 leads naturally to an interpretation that relates the magnitude of the motion generated to integrals of area enclosed by paths in the shape space. We will be most interested in the case where $u(t)$ is cyclic, and so Eq. 3.10 yields a cyclic loop in the shape space about the base point r_0. We include the possibility of a nonzero initial shape, r_0, to allow more flexibility in the modeling of the inputs. The time derivative of r is just $\dot{r} = \epsilon\dot{u}$. A simple perturbation analysis of the momentum equation leads to the following result.

PROPOSITION 3.1 *Given a system starting at rest ($p(0) = 0$), the momentum of the system given an input $r(t)$ of the form given by Eq. 3.10 is*

$$p_a(t) = \epsilon^2\alpha_{aij}(r_0)\int_0^t \dot{u}^i\dot{u}^j d\tau + \epsilon^3\left(\left(\frac{\partial\alpha_{aij}}{\partial r^k} - \beta_{ak}^b\alpha_{bij}\right)\Big|_{r_0}\int_0^t u^k\dot{u}^i\dot{u}^j d\tau + \beta_{ai}^b\alpha_{bjk}(r_0)u^i(t)\int_0^t \dot{u}^j\dot{u}^k d\tau\right) + \ldots.$$

Proof: This result is a straightforward, though extensive, calculation, found by setting $p = p^0 + \epsilon p^1 + \epsilon^2 p^2 + \ldots$, and Taylor expanding α and β about r_0:

$$\begin{aligned}\dot{p} &= \dot{p}^0 + \epsilon\dot{p}^1 + \epsilon^2\dot{p}^2 + \ldots \\ &= 0 + \epsilon\beta(r_0)p^0\dot{u} + \epsilon^2\left(\frac{\partial\beta}{\partial r}\Big|_{r_0} p^0 + \beta(r_0)p^1\dot{u} + \alpha(r_0)\dot{u}\dot{u}\right) + \ldots\end{aligned}$$

Equating terms at each order of ϵ and noting that $p(0) = 0$ leads to the result. ■

Remarks: Proposition 3.1 provides very interesting insights into the generation of momentum for this type of system.

- The same term, α, that is critical for the sufficiency tests of controllability in Prop. 2.4 leads to generation of momentum at the lowest order of ϵ, namely ϵ^2. This makes sense, since α is directly related to the momentum direction as seen through the controllability analysis [20].
- The requirement of Proposition 2.4 that $\alpha_{ii} = 0$ is also important, since the integral in the current proposition related to this motion is of the form $\int(\dot{u}^i)^2$. Since this integral is strictly positive, the momentum generated when $\alpha_{ii} \neq 0$ can only be forced in one direction (positive or negative), dependent on the sign of α_{ii}. This is the case for the roller racer, where α is a nonzero scalar. In this case, the system does not satisfy STLC conditions, since the momentum terms have been found to be uncontrollable [7, 11], but it is provably locally accessible.

- The "off-diagonal" terms of α lead to momentum generation through a mechanism that exactly parallels that found by Bullo and Leonard [4]. As we will see below, the use of in-phase cyclic motions leads to net changes in momentum. We have seen for examples like the snakeboard that having the inputs exactly in-phase produces the largest possible change in the momentum (to first order), as was verified in simulations of the snakeboard in [22].
- Note that the third order term has one integral term that is the same as that found in order ϵ^2, and another of the form

$$\int_0^t u^k \dot{u}^i \dot{u}^j d\tau.$$

This pattern, $\int u^{i_0} \dot{u}^{i_1} \dots \dot{u}^{i_k} d\tau$, is characteristic of the higher order terms as well. It is straightforward to show that for any inputs $u^i = a^i \sin 2\pi m t$ and $u^j = a^j \sin 2\pi n t$, with m and n integers for which $m \neq n$, this integral will be equal to zero after one period.

Proposition 3.1 provides information about the evolution of the momentum. For dynamic nonholonomic systems, this is important, since the momentum is used to drive the system. However, we are generally most interested in the motion of the position variables themselves, since this is the goal of steering. To examine these, we formulate the problem using the single exponential representation (see Magnus [14] and Leonard [12]). In this formulation, we represent the motion in the group variables, $g \in G$, by

$$g(t) = e^{z(t)}, \tag{3.11}$$

and note that the solution to the equation

$$g^{-1}\dot{g} = \epsilon \xi(t) \tag{3.12}$$

is locally given by the infinite series

$$\begin{aligned} z(t) = \epsilon \int_0^t \xi(\tau) d\tau + \frac{\epsilon^2}{2} \int_0^t [\tilde{\xi}(\tau), \xi(\tau)]_{\mathfrak{g}} d\tau + \frac{\epsilon^3}{4} \int_0^t [\int_0^\tau [\tilde{\xi}(\sigma), \xi(\sigma)]_{\mathfrak{g}} d\sigma, \xi(\tau)]_{\mathfrak{g}} d\tau \\ + \frac{\epsilon^3}{12} \int_0^t [\tilde{\xi}(\tau), [\tilde{\xi}(\tau), \xi(\tau)]_{\mathfrak{g}}]_{\mathfrak{g}} d\tau + \dots, \end{aligned} \tag{3.13}$$

where $\tilde{\xi}(t) = \int_0^t \xi(\tau) d\tau$. We can use this expression to derive a local expression for the motion resulting from controlling the shape inputs, $\dot{r}$.

PROPOSITION 3.2 *Given a system initially at rest and inputs of the form specified in Eq. 3.10, the curve in the Lie algebra describing the body velocity satisfies:*

$$\begin{aligned} \xi^\alpha(t) = &- \epsilon \mathbb{A}_i^\alpha(r_0)\dot{u}^i + \epsilon^2 \left(- \left. \frac{\partial \mathbb{A}_i^\alpha}{\partial r^j} \right|_{r_0} \dot{u}^i u^j + (I^{-1}p^2)^\alpha(t) \right) \\ &+ \epsilon^3 \left(- \left. \frac{1}{2} \frac{\partial^2 \mathbb{A}_i^\alpha}{\partial r^j \partial r^k} \right|_{r_0} u^j u^k \dot{u}^i + (I^{-1}p^3)^\alpha + \frac{\partial (I^{-1}p^2)^\alpha}{\partial r^i} u^j \right) + \dots, \end{aligned}$$

where $p(t) = p^0 + \epsilon p^1 + \epsilon^2 p^2 + \dots$ is as defined in Proposition 3.1. Furthermore, the exponential coordinates for the system, for $z(0) = 0$, are given by

$$\begin{aligned} z^\alpha(t) = 0 &+ \epsilon(-\mathbb{A}_i^\alpha u^i|_0^t) + \epsilon^2 \left(-\frac{1}{2} D\mathbb{A}_{ij}^\alpha \int_0^t u^i \dot{u}^j d\tau + (I^{-1})^{\alpha a} \alpha_{aij} \int_0^t \int_0^\tau \dot{u}^i \dot{u}^j ds d\tau \right) \\ &+ \epsilon^3 \left(-\frac{1}{3} \left(\frac{\partial D\mathbb{A}_{ik}^\alpha}{\partial r^j} - [\mathbb{A}_j, D\mathbb{A}_{ij}]_{\mathfrak{g}} \right) \int_0^t u^i u^j \dot{u}^k d\tau + \left(\frac{\partial (I^{-1})^{\alpha a}}{\partial r^j} - [\mathbb{A}_j, (I^{-1})^{\alpha a}]_{\mathfrak{g}} \right) \int_0^t u^j p_a d\tau + (I^{-1})^{\alpha a} \int_0^t p_a^3 d\tau \right) \\ &- \epsilon^2 \left(\frac{1}{2} \frac{\partial \mathbb{A}_i^\alpha}{\partial r^j} (u^i u^j)|_0^t + [\mathbb{A}_i, \mathbb{A}_j]_{\mathfrak{g}}^\alpha u^i(0) u^j|_0^t \right) + \epsilon^3 \mathcal{R}(\mathbb{A}, u) + \dots, \end{aligned} \tag{3.14}$$

where $\mathcal{R}(\mathbb{A}, u)$ represents a function of $\mathbb{A}$ and u that integrates to zero over one period when using cyclic inputs.

Remarks:

- Equation 3.14 is messy at first glance, but upon closer inspection we notice that there are only a few terms that are nonzero after one full period of a cyclic input (regardless of the specific form of the input). At the level of ϵ^2 (first-order bracketing), there are two terms: the first term is related to $D\mathbb{A}$, and represents an area integral, $\int_0^t u^i \dot{u}^j d\tau$. This results from the kinematic effects of the motion, similar to that found by Radford and Burdick for the principal kinematic case [24]. The second term is in the direction $I^{-1}\alpha$, and corresponds to motion driven by the momentum, proportional to $\int_0^t \int_0^\tau \dot{u}^i \dot{u}^j ds d\tau$. Thus, up to second order, the effects of the kinematics and dynamics of the system "cleanly" decouple. For third order, ϵ^3, three terms are present: a "kinematic" term derived exactly from the second-order Lie bracket, $L_X D\mathbb{A}(Y,Z) - [\mathbb{A}(X), D\mathbb{A}Y, Z]_{\mathfrak{g}}$ (see $\mathfrak{h}_3$ in Eq. 2.7); a "dynamic" term related directly to the third order momentum term, p^3; and a cross-coupling term that can be interpreted as a covariant derivative of I^{-1} with respect to the connection, $\mathbb{A}$ (see [24] for more details). This pattern continues to higher orders, though we do not include the details here.

- For small amplitude, short duration cyclic inputs[2] we can move the system to a desired position, g_d, by controlling towards $\xi_d = \log g_d$, where log denotes the inverse operator to exponentiation on the group used in Eq. 3.11.

Before looking at a specific example, we state the result for the case when the system is not initially at rest, but has $p(0) = p_0$.

PROPOSITION 3.3 *Given a system with initial momentum $p(0) = p_0$ and inputs $r(t)$ of the form in Eq. 3.10, the momentum to second order in ϵ is given by*

$$p_a(t) = p_{0a} + \epsilon \beta^b_{ai} p_{0b} u^i|_0^t + \epsilon^2 \left(\left(\frac{\partial \beta^b_{aj}}{\partial r^i} + \beta^c_{aj}\beta^b_{ci} \right) p_{0b} \int_0^t u^i \dot{u}^j d\tau + \alpha_{aij} \int_0^t \dot{u}^i \dot{u}^j d\tau - \alpha^c_{aj}\alpha^b_{ci} p_{0b} u^i(0) u^j|_0^t \right) + \ldots .$$

In this situation, the term that leads to a nonzero net change in momentum is $\int \dot{u}^i \dot{u}^j$, as before. The remaining term is of the form $\int u^i \dot{u}^j$, which is zero over one cycle of inputs if in-phase sinusoidal inputs are used. Thus, we find that even for $p_0 \neq 0$ we can decrease (or otherwise alter) the momentum of the system by using the same type of in-phase inputs as before, but with the sign switched on one of the inputs, i.e., using a 180° change in phase.

4 Algorithms for generating motion

The expressions given in the perturbation expansions of Propositions 3.1–3.3 give us insights into how to generate motion in particular directions. In this section, we investigate how to use these results to provide local changes in the position or momentum using small-scale cyclic inputs. We will see below that these open-loop gaits can be concatenated to move the system towards a desired goal. For a dynamic nonholonomically constrained system, we see that there are two types of cyclic inputs needed to generate changes in the state variables; namely, in-phase cyclic inputs for changing the momentum variables, and out-of-phase cyclic inputs for affecting the system through its kinematic component.

Control of momentum using small amplitude inputs As shown in Propositions 3.1 and 3.3, we can control the momentum to second order using small amplitude cyclic inputs of the form $r(t) = r_0 + \epsilon u(t)$. If we assume that the sufficiency conditions of Proposition 2.4 are satisfied, i.e., that α is onto, but that $\alpha_{ii} = 0$, then each of the momentum directions can be controlled independently by choosing appropriate

[2] We have not attempted to characterize the valid time-scales for these approximations. For more details, see [12, 14].

pairs of inputs, u^i and u^j, and driving them in-phase. For example, if we choose sinusoidal inputs of the form $u^i = a^i \sin 2\pi\omega t$, this leads to a net change of momentum after one cycle equal to

$$\begin{aligned}\Delta p_b &= \alpha_{bij}(r_0) \int_0^1 \dot{u}^i \dot{u}^j \\ &= 2\pi^2\omega^2 \alpha_{bij} a^i a^j.\end{aligned}$$

In fact, a simple check shows that up to order ϵ^2, the in-phase sinusoidal inputs lead to this same change in momentum regardless of the initial momentum, $p(0) = p_0$. In general, one would like to move from $p = p_0$ to p_d, using small magnitude inputs. To achieve this, it is necessary to break the motion up into several repeated cycles, say N cycles. Thus, a suitable method for choosing inputs to control to a desired component of the momentum, $(p_d)_b$, is to choose a magnitude of inputs, a^i and a^j, and a number of cycles, N, such that $2N\pi^2\omega^2\alpha_{aij}a^i a^j = (p_d - p_0)_b$. This will lead to a final momentum $p_b = (p_d)_b + \mathcal{O}(\epsilon^3)$. More generally, similar to Bullo and Leonard [4], one can choose a set of integrally related inputs of the form

$$u^i = \sum_{b=1}^{l} \frac{a_b^i}{b \cdot 2\pi} \sin(b \cdot 2\pi t) \tag{4.15}$$

(summed over b), where l is the number of input directions. In this case, the net effect on the momentum is $\Delta p = \frac{1}{2}(\alpha_{1ij}a_1^i a_1^j, \alpha_{2ij}a_2^i a_2^j, \ldots, \alpha_{lij}a_l^i a_l^j)$. In the case that the sufficient conditions for STLC given in Proposition 2.4 are not satisfied, it may not be possible to *control* the momentum, but it is still possible to compute the effect of cyclic inputs. For example, consider the case where some $\alpha_{bii} \neq 0$, as is true for the roller racer studied in [11]. The use of a single sinusoidal input, $u^i = a^i \sin 2\pi\omega t$, leads to a net momentum change of $\Delta p_b = 2\pi^2\omega^2\alpha_{bii}(a^i)^2$. In this case, Δp_b always has the same sign as α_{bii}, and so can not be controlled arbitrarily. However, we do get an estimate, to third order, of Δp_b.

Generating changes in position and orientation Because of the strong coupling between dynamics and kinematics, the resulting control algorithms for making changes in the group direction are more complex than in either the kinematic or unconstrained cases. The actual computation of the necessary inputs is more realistically done for a particular system, rather than trying to describe explicitly how this should be done in general terms. For this reason, we give here a description of how the computation would work for the first order terms, related to $D\mathbb{A}$, and describe how this relates to higher order terms. Then, we illustrate with the snakeboard example how this can be applied more generally.

For order ϵ in Proposition 3.2, cyclic inputs leave the position, g, unchanged. At order ϵ^2, we see there are two terms. The first term is the same as that found in the principal kinematic case: $D\mathbb{A}$. To generate this direction, we know that we can use inputs that are out of phase, as was suggested based on heuristics by Murray and Sastry [17] and more recently by Radford and Burdick [24]. Thus, picking $u^i = a^i \sin 2\pi\omega t$ and $u^j = a^j \cos 2\pi\omega t$ generates a nonzero net area in the shape space:

$$\int_0^1 u^i \dot{u}^j = -\pi a^i a^j \omega. \tag{4.16}$$

The second term in this equation, $I^{-1}\alpha$, however, is also not zero, since

$$\int_0^1 \int_0^\tau \dot{u}^i \dot{u}^j ds d\tau = \frac{\pi}{2}\omega a^i a^j.$$

Therefore, we must take into account both the effects of the kinematic ($D\mathbb{A}$) and dynamic ($I^{-1}\alpha$) components when planning the control inputs to effect a change in position and orientation.

Finally, we note that for this pairing of inputs $\Delta p = 0$ up to fourth order (and higher), which implies a zero net change in momentum. In fact, this is true more generally for the other out-of-phase input pairs used to generate the higher order Lie derivatives of the connection $\mathbb{A}$. This is because the changes of momentum are related to integrals of the form:

$$\int_0^t u^{i_1} \dot{u}^{i_2} \ldots \dot{u}^{i_k} d\tau,$$

which are zero over one period for any pairing of out-of-phase sinusoidal inputs of the form given above.

Locomotive gaits The controls that arise from this type of analysis have an interesting interpretation in terms of locomotion systems. A very common observation of locomotion is that it is most often generated by *cyclic* shape changes. The motion takes on a characteristic form, called a *gait*, which we can loosely define as a specified cyclic pattern of internal shape changes (inputs) which couple to produce a net motion.

For each species, there usually exist at most a handful of gaits, often tailored for specific needs or environments. For instance, a human will walk or run, depending on the desired speed, but may also hop or skip (though these two gaits do not seem to serve any evolutionary function). In the present case, we notice that the coupling of cyclic inputs gives rise to nonzero integrals of area and momentum in particular directions related to the derivatives of the connection and the momentum terms. In this context, we can view the in-phase gaits as cyclic patterns that lead to a net generation of momentum, while the out-of-phase gaits contribute to the kinematic mechanism for generating motion (as with traditional kinematic nonholonomic systems). These ideas are illustrated by the following example.

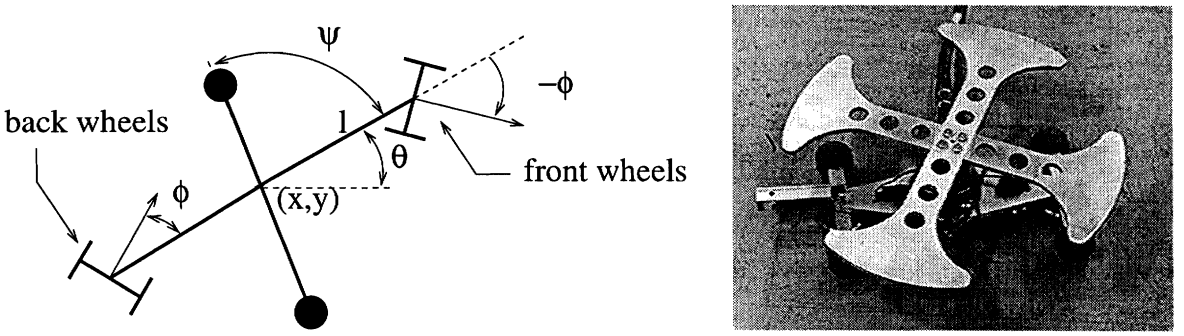

Figure 1: The simplified model of the *Snakeboard*, along with a demo prototype.

The Snakeboard Example The *Snakeboard* is a commercial variant of the skateboard that allows for independent rotation of the wheel trucks. The simplified model of the *Snakeboard* (referred to as the snakeboard model) is shown in Figure 1, along with a robotic version that was constructed in order to verify theoretical simulations (also shown in Figure 1). We briefly recall the description of the snakeboard as developed in [22, 23]. The basic premise is that by properly coupling the rotating flywheel (modeling the swinging of a human torso) with the turning of the wheels, one can generate a variety of motions, including a forward serpentine-like gait reminiscent of a snake. As a mechanical system the snakeboard has a configuration manifold given by $Q = SE(2) \times \mathbb{S} \times \mathbb{S}$. As coordinates for Q we shall use $(x, y, \theta, \psi, \phi)$ where (x, y, θ) describes the position of the board with respect to a reference frame, ψ is the angle of the rotor with respect to the board, and ϕ denotes the angle of the back wheel truck with respect to the board (and the opposite of the angle of the front wheels, which is assumed to move through equal and opposite rotations). Note that the wheels themselves are allowed to spin freely, just as with a traditional skateboard.

The wheels of the snakeboard are assumed to roll without lateral sliding. This leads to nonholonomic constraints of the form

$$-\sin(\phi+\theta)\dot{x} + \cos(\phi+\theta)\dot{y} - l\cos(\phi)\dot{\theta} = 0 \tag{4.17}$$

$$\sin(\phi-\theta)\dot{x} + \cos(\phi-\theta)\dot{y} + l\cos(\phi)\dot{\theta} = 0. \tag{4.18}$$

The nonholonomic momentum, p, is defined along directions which are tangent to the fiber and lie in the constraint distribution (i.e, satisfy the constraints). For the snakeboard, the momentum is proportional to the instantaneous angular momentum of the snakeboard about the center of rotation defined by the two wheel constraints. In terms of the above quantities, this is

$$p = -ml\cos^2\phi(\dot{x}\cos\theta + \dot{y}\sin\theta) + \sin 2\phi\left((J + J_r + 2J_w)\dot{\theta} + J_r\dot{\psi} + 2J_w\dot{\phi}\right). \tag{4.19}$$

Following along the lines of Ostrowski et al. [23], we use a non-dimensional set of variables, normalized by the board length and rotor momentum, so that the fiber equations reduce to the form of Eqs. 2.2 with

$$\mathbb{A} = \begin{pmatrix} -\hat{J}\sin\phi\cos\phi & 0 \\ 0 & 0 \\ \hat{J}\sin^2\phi & 0 \end{pmatrix} \quad \text{and} \quad I^{-1} = \begin{pmatrix} -\frac{\hat{J}}{2} \\ 0 \\ \frac{\hat{J}}{2}\tan\phi \end{pmatrix},$$

and $\hat{J}$ a non-dimensional ratio between the rotor inertia and the board inertia. The generalized momentum equation, Eq. 2.3, is then given by

$$\dot{p} = 2\cos^2\phi\,\dot{\phi}\dot{\psi} - \tan\phi\,\dot{\phi}p.$$

Next, we can explore the relationship between the integrals of Proposition 3.1 and the appearance of cyclic inputs.

The α term in the momentum equation leads to motion in the momentum direction for

$$\psi(t) = \frac{\pi}{2} - a_\psi \sin 2\pi t$$
$$\phi(t) = a_\phi \sin 2\pi t.$$

A set of simulations for this pair of inputs are presented in Figure 2. The left-hand plot in this figure shows the time evolution of the momentum. For each set of inputs, the predicted value of the momentum after one cycle is shown as a horizontal line. This is based on the calculation:

$$\Delta p = \alpha_{aij}(r_0)\int_0^1 u^i u^j d\tau = 2\int_0^1 \dot{\phi}\dot{\psi}d\tau$$
$$= -2(2\pi)^2 a_\phi a_\psi \int_0^1 \cos^2 2\pi\tau\, d\tau = -8\pi^2 a_\phi a_\psi/2 = -4\pi^2 a_\phi a_\phi.$$

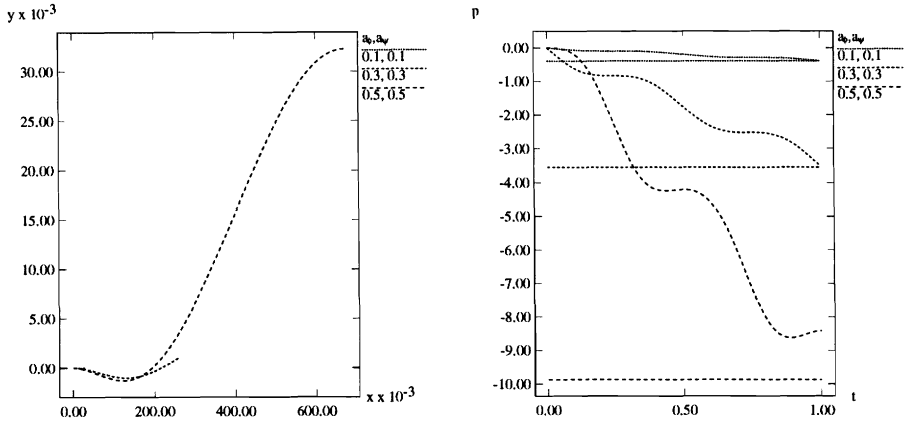

Figure 2: Forward motion during momentum generation, and momentum versus time. The approximated value of the momentum is shown in horizontal lines

It should be noted that the prediction for the small inputs works quite well up to an amplitude (in radians) of approximately $a_\phi = a_\psi = 0.3$. Larger amplitude inputs ($a_\phi = a_\psi = 0.5$) are also plotted to show that while some discrepancy does exist between the predicted and simulated final values of momentum, the error is relatively still relatively small (on the order of 15%).

Each of the other brackets has a related cyclic pattern. For the 1:1 gait, we examine the terms $D\mathbb{A}$:

$$(D\mathbb{A})_{\phi\psi} = \left(-\hat{J}\cos 2\phi, \quad 0, \quad \hat{J}\sin 2\phi\right)^T \in \mathfrak{h}_2, \tag{4.20}$$

where $(D\mathbb{A})_{\phi\psi}|_0 = (-\hat{J}, 0, 0)^T$, and $I^{-1}\alpha$:

$$(I^{-1}\alpha)_{\phi\psi} = \frac{1}{2}(-\hat{J}\cos^2\phi, 0, \hat{J}\sin\phi\cos\phi)^T \in \mathfrak{h}_2, \tag{4.21}$$

with $(I^{-1}\alpha)_{\phi\psi}|_0 = -\frac{1}{2}(\hat{J}, 0, 0)^T$.

As mentioned above, we see that the use of out-of-phase gaits,

$$\begin{aligned}\psi(t) &= \pi/2 + a_\psi \sin 2\pi t \\ \phi(t) &= a_\phi \cos \omega t\end{aligned}$$

leads, up to second order, to a net motion of

$$\begin{aligned}\Delta x &= -\frac{1}{2}D\mathbb{A}^x_{\phi\psi}\int_0^1 \phi\dot{\psi}dt - \frac{1}{2}D\mathbb{A}^x_{\psi\phi}\int_0^1 \psi\dot{\phi}dt + 2(I^{-1}\alpha)^x_{\phi\psi}\int_0^1\int_0^t \dot{\phi}\dot{\psi}d\tau dt \\ &= \frac{1}{2}\hat{J}\int_0^1 (\phi\dot{\psi} - \psi\dot{\phi} + 2\dot{\phi}\dot{\psi})dt \\ &= \frac{1}{2}\hat{J}(\pi a_\phi a_\psi + \pi a_\psi a_\phi - \pi a_\phi a_\psi) \\ &= \frac{3\pi}{2}\hat{J}a_\phi a_\psi.\end{aligned}$$

Note that we have written the net motion in terms of x, even though Proposition 3.2 gives the motion in terms of exponential coordinates, $z(t)$. This is because $z(1)$ is of the form $(z_1, 0, 0)$, for which $(x, y, \theta)(1) = e^{z(1)} = (z_1, 0, 0)$. This forward gait has the kinematic and dynamic effects on the forward motion working together, that is, with the same sign. Interestingly, if we choose the opposite choice for out-of-phase inputs,

$$\begin{aligned}\psi(t) &= \pi/2 + a_\psi \cos 2\pi t \\ \phi(t) &= a_\phi \sin \omega t,\end{aligned}$$

we find that the kinematic and dynamic effects work opposite each other and give

$$\Delta x = -\pi\hat{J}a_\phi a_\psi + \frac{\pi}{2}\hat{J}a_\phi a_\psi = -\frac{\pi}{2}\hat{J}a_\phi a_\psi.$$

Also important is that there is no net effect on the momentum:

$$\Delta p = 2\alpha_{\phi\psi}\int_0^1 \dot{\phi}(t)\dot{\psi}(t)dt = 0, \tag{4.22}$$

so that the motion generation and momentum generation steps are decoupled.

The forward motion is plotted in Figure 3, where we have indicated with vertical bars the value predicted by the approximation. We see that for small inputs, the error is negligible, and even for inputs where $a_\phi = a_\psi = 0.5$, the error is still on the order of 10%. Also shown are two gaits which have the same area in shape space, but generated through different magnitudes of the individual shape variables. These demonstrate that while the area rule may work well for smaller inputs, it does not necessarily hold true for larger magnitude inputs.

It is also interesting to note that this gait was never fully examined in the original work with the snakeboard. The other gaits have already been studied in previous works [21, 22], and so are not examined in detail here. However, we see that the higher derivatives of the connection lead to terms of the form

$$\left(-2\hat{J}\sin 2\phi, \quad 0, \quad -2\hat{J}\cos 2\phi\right)^T \in \mathfrak{h}_3, \quad \text{and} \quad \left(0, \quad 2\hat{J}^2\cos 2\phi, \quad 0\right)^T \in \mathfrak{h}_5,$$

corresponding to motion in the θ and y directions, respectively. As described in [19, 21], these two motions can be generated using $\phi = a_\phi \sin 2\pi t$ with $\psi^\theta = a_\psi \sin 4\pi t$ and $\psi^y = a_\psi \sin 3\pi t$, respectively. For a little better understanding of why these inputs are the appropriate choices, we can look at the integrals that

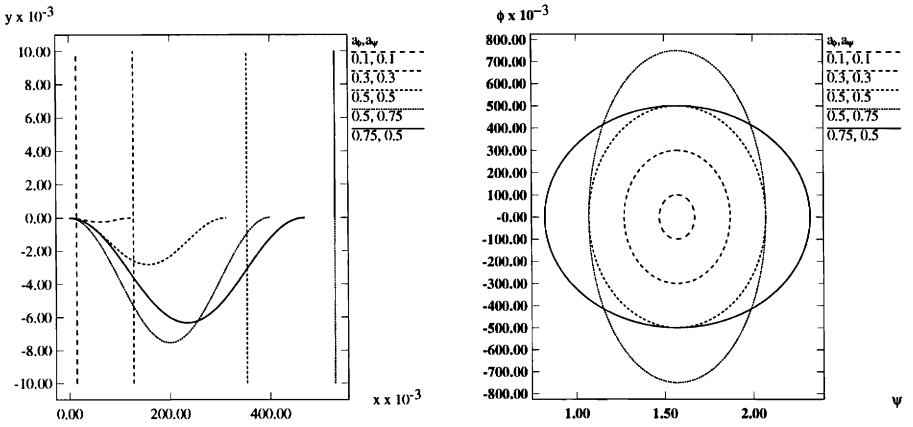

Figure 3: Forward motion with net zero momentum, along with the inputs

generate them, as in Proposition 3.2. For the θ direction, the integral related to the kinematic component of the motion is given in Eq. 3.14 as

$$\int_0^t u^i u^j \dot{u}^k d\tau. \tag{4.23}$$

For the given set of inputs, this evaluates to $-\pi a_\phi^2 a_\psi$.

5 Steering for dynamic nonholonomic systems

In the previous section, we derived results for controlling the individual state variables of a dynamic nonholonomic system. In this section, we explore the use of steering algorithms for mobile robots in controlling dynamic systems. While the results are fairly straightforward applications of the above algorithms to motion planning, they are useful because they allow us to extend our domain of control to a more non-local region of the state-space. In other words, instead of being forced to patch together purely local results, one can determine motion plans for more non-local control goals. This has a similar feel to the steering of a mobile robot or an airplane, where one pieces together very simple trajectories to generate global (not STLC) controllability. Certainly, it is possible to generate more sophisticated algorithms for determining the motion plans for moving from point to point. One can even include more involved requirements on the motion planner, such as obstacle avoidance and minimization of energy [23]. However, we illustrate one way in which this type of motion planning can be utilized in the case of steering systems.

First, recall the results of Dubins [6] for kinematic mobile robots. It was shown that the optimal paths for a car steering in $SE(2)$ with limits on the turning radius were given by two circular arcs (one arc tangent to each of the initial and final headings) connected by a straight line segment. In this section we utilize similar ideas in generating steering algorithms for dynamic nonholonomic systems. Although we haven't done it here, one could also extend these algorithms to take advantage of systems that are STLC, for which the results of Reeds and Shepp [25] on cars that can move forwards and backwards would apply.

We note that we restrict our attention in this section to "car-like" systems that move in $SE(2)$ and are "steerable" using one or more of the internal state variables. Steerable in this context is loosely taken to mean that the system evolves along a special (1-D) subset of Lie algebra directions. To be slightly more precise, we assume to be *steerable* any system for which we can use the shape variables to generate body velocities (c.f., Eq. 2.2) of the form

$$\xi = g^{-1}\dot{g} = (v, 0, v/R)^T \tag{5.24}$$

where v is the forward velocity, and $0 < R < R_{\max}$ represents the turning radius (and $R_{\max}$ a turning radius constraint). This effectively means that we can create generalized circles, using either the shape inputs directly, or through the momentum of the system.

For the snakeboard example, steering comes from controlling ϕ as it affects the term $I^{-1} = (-\hat{J}/2, 0, \hat{J}\tan\phi/2)^T$. Extensions to other Lie groups are certainly possible, by considering geodesics that can be generated using whatever allowable steering inputs are given. See, for example, Zefran and Kumar [28]. We also mention that these steering results can be useful for other types of wheeled vehicles, such as bikes, and may also be extended beyond wheeled systems, for example, for use in controlling fish- or eel-like swimming.

The steering algorithm, then, requires two simple steps:

1. **Momentum generation:** As shown above in Proposition 3.1, momentum can be generated by the simple mechanism of in-phase inputs. For steering in $SE(2)$, the momentum is assumed to be one-dimensional. Thus, this step takes the system from $(0,0,0,0)$ to $(x_1, y_1, \theta_1, p_d)$, for some desired momentum, p_d.

2. **Steering to the desired final point:** Once the momentum has been built up, the system can be steered along any desired path (by assumption). This involves computing the necessary arcs and line segments to connect (x_1, y_1, θ_1) to (x_d, y_d, θ_d). In order to setup for the final step of reducing the momentum, steer the system to the point $(x_d - x_1, y_d - y_1, \theta_d - \theta_1)$. Assuming there is no damping in the system, the final point is reached with a momentum of p_d.[3] This assumes that the same change in position will result during the momentum reduction phase.

3. **Zero the momentum:** Execute the opposite maneuver to that chosen in Step 1 in order to reduce the momentum to zero.

4. **Cancel errors in position and momentum:** Since the above analysis is just an approximation, it is expected that the final state after Step 2 will not be exactly $(x_d, y_d, \theta_d, 0)$, but instead some $(\hat{x}, \hat{y}, \hat{\theta}, \hat{p})$. The final step, then, requires using the simple gait pairs described in Section 4 to make this correction, starting with zeroing the momentum, $\hat{p}$. Step 3 is then repeated until the position, orientation, and momentum are all within some desired error.

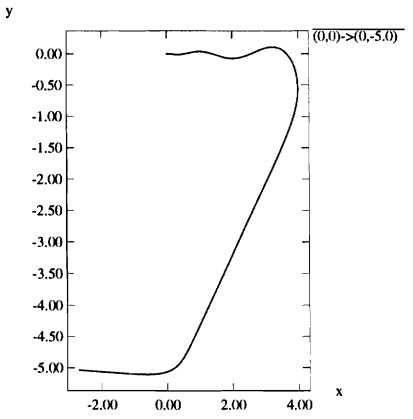

Figure 4: Steering using only Steps 1 and 2

If one is interested strictly in configuration controllability (moving between configurations only) [5], then the motion can be achieved using Steps 1 and 2. To see how these two steps could be implemented, we

[3] In the presence of dissipation, one needs to add a small amplitude in-phase oscillation to the steering commands. This will allow the system to maintain an average steady-state nonzero momentum. Since the effects of these commands is essentially decoupled, the steering follows largely unaffected. Initial investigations of this for the roller racer can be found in [7].

provide in Figure 4 a sample plot of motion control for the snakeboard moving from $(x, y, \theta) = (0, 0, 0)$ to $(0, -5.0, 0)$. Notice that in this plot the system first builds up momentum using a serpentine-like snakeboard "drive" gait in Step 1, and then executes a turn, a straight coast, and a final turn in Step 2. The turns are executed at the minimum turning radius, which we have constrained to be determined by $\phi_{max} = \pm 1.0$rad. We have also extended the run-time of the simulation past the point $(0, -5, 0)$, in order to show the nonzero final momentum, as well as the effect of a small error in the calculation of the final orientation.

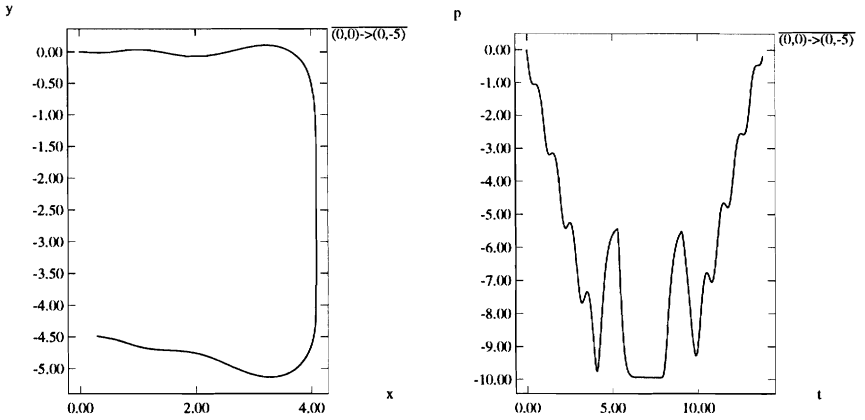

Figure 5: Steering using the full algorithm, along with associated changes in momentum

Using the first three steps of the algorithm allows us to converge more closely to the desired point. A simulation of this maneuver is given in Figure 5, along with a plot of the momentum as a function of time. For this simulation, we execute the same set of input motions at the end of the trajectory as was used at the start, except that the wheel angles are turned through opposite rotations of those used originally. These in-phase cyclic gaits used to build up and to reduce momentum at the start and finish of the motion are run for the same length of time. As is clear from the graph, this results in a near exact cancelation of the momentum that was built up during the initial motions.

Obviously, each of these algorithms is only approximate— more involved feedback laws could be developed along the lines of [4]. However, one really only needs to move the system close to the desired goal, at which point the simple coupled inputs (gaits) given in Section 4 can be used to control the motion (Step 4).

6 Conclusions and Future Work

This paper establishes easily implemented open-loop control algorithms for mechanical systems with Lie group symmetries and external nonholonomic constraints. We have developed two methods for doing so, with distinctly different flavors. The first uses small amplitude sinusoidal inputs to make small, open-loop steps in desired directions. This provides an extension to similar previous work by Murray and Sastry on steering kinematic systems, and Bullo and Leonard for unconstrained mechanical systems. We have shown through simulation that for small magnitude inputs the predicted approximation matches quite well with results from simulation. This algorithm provides a mechanism by which a controller can be designed to iteratively move towards a desired goal by using repeated cyclic loops of the shape inputs. It also specifies a method for controlling the momentum of the system, starting from any given momentum.

The second set of procedures follows along the lines of motion planning algorithms developed by Dubins for controlling kinematic wheeled vehicles. It provides a more non-local method for generating changes in position and orientation for dynamic steering problems by using shape inputs to control steering. The main idea presented here is that a dynamic steering problem can generally be reduced to an equivalent kinematic steering problem where one must add initial and final steps in the algorithm to achieve the necessary motion (i.e., momentum). The steering is then generated using the internal shape variables.

The present study has certainly raised additional issues that merit further research.

- We have restricted our attention to systems where the quadratic term in the momentum equation, γ, is identically zero. This is motivated by the form of the equations found in dynamic steering problems to date. Further examination of the mechanical structure of the equations could lead to interesting insights into how to generalize this work.
- The perturbation techniques presented here are certainly not new. However, they do provide a powerful tool for examining the effect of cyclic inputs for general systems of the form $\dot{x} = f(x, u)$.
- We have examined only idealized models, where there is no friction or other disturbance in the system. Generally, viscous friction enters as a linear term in the momentum equation [7]. We have seen in simulation, however, that there are simple modifications to the above algorithms that can accommodate such friction terms. We are also currently working on basic extensions to these ideas that will work in related systems such as underwater and aerial robotic systems.

Acknowledgments

The author would like to thank Francesco Bullo, Naomi Leonard, and Jim Radford for their many useful discussions on the use of averaging and perturbation theory in constructive controls. Thanks also to Joel Burdick, Vijay Kumar, Andrew Lewis, Jerry Marsden, and Richard Murray for their continued support and advice. I would also like to acknowledge the support of NSF grant IRI-9711834, and pay a special thanks to Dr. Howard Moraff, who is retiring from NSF, for supporting my research program.

References

[1] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications.* Springer-Verlag, New York, 1988.

[2] A. M. Bloch, P. S. Krishnaprasad, J. E. Marsden, and R. M. Murray. Nonholonomic mechanical systems with symmetry. *Archive for Rational Mechanics and Analysis*, 136:21–99, December 1996.

[3] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch. Control and stabilization of nonholonomic dynamic systems. *IEEE Trans. on Automatic Control*, 37(11):1746–1757, 1992.

[4] F. Bullo, N. E. Leonard, and A. D. Lewis. Controllability and motion algorithms for underactuated Lagrangian systems on Lie groups. Submitted to IEEE *Transactions on Automatic Control*, February 1998.

[5] F. Bullo and A. D. Lewis. Configuration controllability for mechanical systems on Lie groups. In *Symposium on Mathematical Theory of Networks and Systems*, St. Louis, June 1996.

[6] L. E. Dubins. On curves of minimal length with a constraint on average curvature and with presribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79:497–516, 1957.

[7] E. N. Edgar. Dissipation effects in the roller racer with investigations into motion planning. Master's thesis, Univ. of Pennsylvania, Philadelphia, December 1997.

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

[9] H. Khalil. *Nonlinear Systems.* Macmillan Publishing Co., 1992.

[10] P. S. Krishnaprasad. Geometric phases and optimal reconfiguration for multibody systems. In *Proc. of the 1990 American Control Conference*, pages 2440–2444, Philadelphia, 1990. American Automatic Control Council.

[11] P. S. Krishnaprasad and D. P. Tsakiris. G-snakes: Nonholonomic kinematic chains on Lie groups. In *Proc.* 33^{rd} *IEEE Conf. on Decision and Control*, pages 2955–2960, Lake Buena Vista, FL, December 1994.

[12] N. E. Leonard and P. S. Krishnaprasad. Motion control of drift-free, left-invariant systems on Lie groups. *IEEE Trans. on Automatic Control*, 40(9):1539–1554, September 1995.

[13] A. D. Lewis. When is a mechanical control system kinematic? In *Proc. IEEE Conf. on Decision and Control*, December 1999. To appear.

[14] W. Magnus. On the exponential solution of differential equations for a linear operator. *Communications on Pure and Applied Mathematics*, VII:649–673, 1954.

[15] J. E. Marsden and J. Ostrowski. Symmetries in motion: Geometric foundations of motion control. To appear in *Nonlinear Science Today*, January 1998. Available via http://www.cis.upenn.edu/~jpo/papers.html.

[16] R. Montgomery. Nonholonomic control and gauge theory. In Z. Li and J. F. Canny, editors, *Nonholonomic Motion Planning*, pages 343–378. Kluwer, 1993.

[17] R. M. Murray and S. S. Sastry. Nonholonomic motion planning: Steering using sinusoids. *IEEE Transactions on Automatic Control*, 38(5):700–716, 1993.

[18] H. Nijmeijer and A. J. van der Schaft. *Nonlinear Dynamical Control Systems*. Springer-Verlag, New York, 1990.

[19] J. P. Ostrowski. *The Mechanics and Control of Undulatory Robotic Locomotion*. Ph.D. thesis, California Institute of Technology, Pasadena, CA, 1995. Available electronically at http://www.cis.upenn.edu/~jpo/papers.html.

[20] J. P. Ostrowski and J. W. Burdick. Controllability tests for mechanical systems with symmetries and constraints. *J. Applied Mathematics and Computer Science*, 7(2):305–331, June 1997.

[21] J. P. Ostrowski and J. W. Burdick. The geometric mechanics of undulatory robotic locomotion. *International Journal of Robotics Research*, 17(7):683–702, July 1998.

[22] J. P. Ostrowski, A. D. Lewis, R. M. Murray, and J. W. Burdick. Nonholonomic mechanics and locomotion: The snakeboard example. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2391–7, San Diego, CA, May 1994.

[23] J. Ostrowski, J. P. Desai, and V. Kumar. Optimal gait selection for nonholonomic locomotion systems. Submitted to the *International Journal of Robotics Research*, July 1998.

[24] J. Radford and J. Burdick. Local motion planning for nonholonomic control systems evolving on principal bundles. Submitted to *Conf. Mathematical Theory of Networks and Systems*.

[25] J. A. Reeds and L. A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990.

[26] H. J. Sussman. A general theorem on local controllability. *SIAM Journal on Control and Optimization*, 25(1):158–194, January 1987.

[27] G. C. Walsh and S. S. Sastry. On reorienting linked rigid bodies using internal motions. *IEEE Transactions on Robotics and Automation*, 11(1):139–146, 1995.

[28] M. Zefran, V. Kumar, and C. Croke. On the generation of smooth three-dimensional rigid body motions. *IEEE Trans. Robotics and Automation*, 1995. (Under review).