Approximate Steering of a Unicycle Under Bounded Model Perturbation Using Ensemble Control

Aaron Becker, *Student Member, IEEE,* and Timothy Bretl, *Member, IEEE*

***Abstract*—This paper considers the problem of steering a nonholonomic unicycle despite model perturbation that scales both the forward speed and the turning rate by an unknown but bounded constant. We model the unicycle as an ensemble control system, show that this system is ensemble controllable, and derive an approximate steering algorithm that brings the unicycle to within an arbitrarily small neighborhood of any given Cartesian position. We apply our work to a differential-drive robot with unknown but bounded wheel radius, and validate our approach with hardware experiments.**

***Index Terms*—motion planning, ensemble control theory, bounded uncertainty**

I. INTRODUCTION

IN this paper we apply the framework of ensemble control theory [1]–[8] to derive an approximate steering algorithm for a nonholonomic unicycle in the presence of model perturbation that scales both the forward speed and the turning rate by an unknown but bounded constant. The basic idea, similar to early work on sensorless manipulation [9], is to maintain the set of all possible configurations of the unicycle and to select inputs that reduce the size of this set and drive it toward some goal configuration. The key insight is that the evolution of this set can be described by a family of control systems that depend continuously on the unknown constant. Ensemble control theory provides conditions under which it is possible to steer this entire family to a neighborhood of the goal configuration with a single open-loop input trajectory. These conditions mimic classical tests of nonlinear controllability like the Lie algebra rank condition [10] but involve approximations by repeated Lie bracketing that are reminiscent of seminal work on steering nonholonomic systems by Lafferriere and Sussman [11].

In particular, consider a single unicycle that rolls without slipping. We describe its configuration by $q = (x, y, \theta)$ and its configuration space by $\mathcal{Q} = \mathbb{R}^2 \times \mathbb{S}^1$. The control inputs are the forward speed u_1 and the turning rate u_2. We restrict $(u_1, u_2) \in \mathcal{U}$ for some constraint set $\mathcal{U} \subset \mathbb{R}^2$, where we assume that $\mathcal{U}$ is symmetric with respect to the origin and that the affine hull of $\mathcal{U}$ is $\mathbb{R}^2$. Corresponding to these inputs, we define vector fields $g_1, g_2 \colon \mathcal{Q} \to T_q\mathcal{Q}$ by

$$g_1(q) = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \qquad\qquad g_2(q) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

A. Becker is with the Department of Electrical and Computer Engineering and T. Bretl is with the Department of Aerospace Engineering at the University of Illinois at Urbana-Champaign, Urbana, IL, 61801 USA, email: {abecker5,tbretl}@illinois.edu.

and write the kinematics of the unicycle in the standard form

$$\dot{q}(t) = g_1(q(t))u_1(t) + g_2(q(t))u_2(t). \tag{1}$$

Given $q_{\text{start}}, q_{\text{goal}} \in \mathcal{Q}$ and $\mu > 0$, the approximate steering problem is to find open-loop inputs

$$(u_1(t), u_2(t)) : [0, T] \to \mathcal{U}$$

that result in $q(0) = q_{\text{start}}$ and $\|q(T) - q_{\text{goal}}\| \leq \mu$ for free final time T, where $\|\cdot\|$ is a suitable norm on $\mathcal{Q}$. If such inputs always exist then we say that (1) is *approximately controllable*—and indeed they do, since g_1, g_2, and the Lie bracket $[g_1, g_2]$ span the tangent space $T_q\mathcal{Q}$ everywhere.

We will solve this same approximate steering problem, but under model perturbation that scales both the forward speed u_1 and the turning rate u_2 by some unknown, bounded constant. The resulting kinematics have the form

$$\dot{q}(t) = \epsilon \left(g_1(q(t))u_1(t) + g_2(q(t))u_2(t) \right), \tag{2}$$

where $\epsilon \in [1-\delta, 1+\delta]$ for some $0 \leq \delta < 1$. Rather than try to steer one unicycle governed by (2)—where ϵ is unknown—our approach is to steer an uncountably infinite collection of unicycles parameterized by ϵ, each one governed by

$$\dot{q}(t, \epsilon) = \epsilon\big(g_1\left(q(t,\epsilon)\right) u_1(t) + g_2\left(q(t,\epsilon)\right) u_2(t)\big). \tag{3}$$

Following the terminology introduced by [1]–[8], we call this fictitious collection of unicycles an *ensemble* and call (3) an *ensemble control system*. The idea is that if we can find open-loop inputs $u_1(t)$ and $u_2(t)$ that result in $q(0, \epsilon) = q_{\text{start}}$ and $\|q(T, \epsilon) - q_{\text{goal}}\| \leq \mu$ for all $\epsilon \in [1-\delta, 1+\delta]$, then we can certainly guarantee that the actual unicycle, which corresponds to one particular value ϵ^* of ϵ, will satisfy $\|q(T, \epsilon^*) - q_{\text{goal}}\| \leq \mu$. If such inputs always exist then we say that (3) is *ensemble controllable*, interpreted as being approximately controllable on the function space $L_2\left([1-\delta, 1+\delta], \mathcal{Q}\right)$. We will in fact show that (3) is not ensemble controllable, but will proceed to derive a reduced subsystem that is. Our proof will depend on being able to approximate arbitrary elements of the tangent space to $L_2\left([1-\delta, 1+\delta], \mathcal{Q}\right)$, capturing the essence of classical tests like the Lie algebra rank condition. Solving the approximate steering problem with respect to the subsystem will produce inputs that reach an arbitrarily small neighborhood of any Cartesian position, but not of any heading.

We will apply our work to a differential-drive robot with unknown but bounded wheel radius, showing that (2) is an appropriate model and validating our approach to approximate steering with hardware experiments.

However, let us be clear—in its current form, this application of our work is not "practical" and does not improve upon existing methods of motion planning and control for mobile robots. In particular, these robots typically have both proprioceptive (e.g., odometry) and exteroceptive (e.g., sonar, laser, vision) sensors. With state estimates that come from these sensors, it is easy to build a feedback controller that guarantees exact asymptotic convergence to any given Cartesian position under the same type of model perturbation that we consider [12, Chap. 11.6.2]. It is just as easy to build a robust feedback controller that regulates posture and not just Cartesian position [13]. Methods like these extend to a broader class of model perturbation (e.g., scaling forward speed and turning rate by different amounts) and to other types of uncertainty. There is also an enormous literature on odometry calibration for wheeled mobile robots to reduce model perturbation [14], [15], from offline approaches like "UMBmark" [16] to approaches that are online [17] and even simultaneous with localization [18]. These citations represent only a fraction of prior work on calibration and robust feedback control, all of which is more effective than what we propose when sensors are available.

Instead, we use the differential-drive robot as a hardware platform because the application of ensemble control theory to this system is easy to understand and leads to results that readers may find surprising. For example, our approximate steering algorithm—derived for an infinite-dimensional family of control systems and not just for a single unicycle—ultimately requires solving only one set of linear equations, which can be precomputed in closed form. Similarly, the formulation of these linear equations relies on series expansions that make explicit the trade-off between the cost and complexity of the resulting input trajectory and the extent to which this input trajectory is robust to model perturbation. Finally, the fact that inputs executed in open-loop will bring a real mobile robot to a neighborhood of the same Cartesian position regardless of wheel radius is something that we did not initially think possible (see the video attachment).

We hope that these results stimulate interest and provoke a new line of inquiry that may lead to practical application in robotics. Our own interest in ensemble control, for example, is largely motivated by potential application to grasping and sensorless manipulation. Consider the plate-ball problem introduced by Brockett and Dai [19], a case study of rolling bodies in contact [20]–[22]. If the ball has unknown but bounded radius, the reader may verify—using a method of analysis similar to the one we describe in this paper—that the resulting system is ensemble controllable. This result hints at a new approach to robust manipulation of so-called "toleranced parts" [23], [24], an ongoing problem in automated assembly and industrial parts handling. A considerable amount of work remains to be done, however, before ideas like this one find their way into practice.

The remainder of this paper is organized as follows. We begin in Section II with a brief review of ensemble control theory and other related work. In Section III, we proceed to show that (3) is not ensemble controllable but that a reduced subsystem is. Based on this result, we derive an approximate steering algorithm in Section IV that brings the unicycle to within an arbitrarily small neighborhood of any given Cartesian position, regardless of ϵ. Finally, in Section V, we validate our approach in experiments with a differential-drive robot that has unknown but bounded wheel radius.

Note that a preliminary conference version of this paper has appeared [25], but that the approximate steering algorithm we present here is quite different.

II. Related Work

A. Ensemble Control

Ensemble control, as presented in [1]–[8], extends the theory of nonlinear controllability from finite-dimensional systems, for example of the form (1), to a particular class of infinite-dimensional systems characterized by a dispersion parameter, for example of the form (3) where this parameter is ϵ. The fact that standard controllability theorems rely on checking a rank condition is a clue that such an extension might be necessary. Chow's theorem, for instance, implies that (1) is small-time locally controllable—hence, approximately controllable—because the Lie algebra generated by g_1 and g_2 has rank 3 everywhere, equal to the dimension of $T_q\mathcal{Q}$ (e.g., see [26]). Both $L_2\left([1-\delta, 1+\delta], \mathcal{Q}\right)$ and its tangent space have infinite dimension, so we will never accumulate enough vector fields to satisfy this same rank condition for (3). However, we note that the rank condition is only used to guarantee that it is possible to approximate motion in any direction we like. For (1), we see that g_1, g_2, and

$$[g_1, g_2] = \frac{\partial g_2}{\partial q} g_1 - \frac{\partial g_1}{\partial q} g_2$$

are linearly independent and span the tangent space $T_q\mathcal{Q}$ at every configuration $q \in \mathcal{Q}$, so any element of $T_q\mathcal{Q}$ can be approximated by rapid switching between inputs. For (3), it is possible to arrive at a similar result. In particular, we will take the same basic approach as in [7], using repeated bracketing to get higher-order powers of ϵ and then using polynomial approximation to construct arbitrary vector flows. Systems like ours are ignored by [7] after noting that

$$\dot{q}(t, \epsilon) = \epsilon \sum_{i=1}^{m} g_i\left(q(t, \epsilon)\right) u_i(t)$$

is not ensemble controllable if $g_1, \ldots, g_m$ generate a nilpotent Lie algebra. We will indeed show that (3) is not ensemble controllable in Section III, but will then proceed to derive a reduced subsystem that is ensemble controllable.

The origins of this approach are within the physics community. In this context, an "ensemble" is a very large collection of identical or nearly identical molecules, atoms, or elementary particles, and the goal of "ensemble control" is to manipulate the average properties of such an ensemble. Early work in this area was done, for example, by Simon van der Meer, who won the 1984 Nobel prize in physics for controlling the density at which circulating protons are packed in an accelerator using applied magnetic fields [27]. The more recent work of Brockett, Khaneja, and Li has found primary application so far to quantum systems, for example manipulating nuclear spins in Nuclear Magnetic Resonance (NMR) spectroscopy [1]–[8]. Robotics researchers are also beginning to adopt the term "ensemble," for example in the context of multi-robot formations [28] and artificial muscle actuators [29], but the formal methodology of ensemble control has yet to be applied. Other approaches to dealing with infinite-dimensional systems—such as taking advantage of differential flatness—have been developed in parallel, as in [30]. The main tool used in this other work is functional analysis, which has recently started to inform ongoing work in ensemble control [31].

B. Robust Control

Robust control provides a framework for the design of feedback policies that compensate for model perturbation. Consider the dynamic system

$$\begin{aligned} \dot{x} &= f(x, u, \epsilon) \\ y &= h(x, u, \epsilon), \end{aligned}$$

where x is the state, u is the input, y is the measurement, and ϵ is an unknown but bounded parameter. We are free to define an equivalent system

$$\begin{aligned} \dot{x} &= Ax + Bu + w_1 \\ y &= Cx + Du + w_2, \end{aligned}$$

for example by linearization, that pushes all model perturbation and nonlinearity into the mapping

$$(x, u, \epsilon) \mapsto (w_1, w_2) \,.$$

The idea is then to replace this one unknown mapping by a set of known linear mappings that capture all possible input/output behavior. This approach has been a topic of study for over fifty years—a modern reference is the book [32]. Although our own work has much the same flavor, robust control theory is primarily focused on the problem of closed-loop stabilization with feedback, whereas we focus on the problem of open-loop steering in the absence of sensor measurements. We emphasize again that robust feedback control is, in general, much more effective than what we propose when sensors are available, as they typically are for mobile robots (see Section I). We also note previous work on robust feedforward control [33], on robust control using series expansions similar to what we will describe in Section IV [34], and on the relationship between ensemble control and robust control [35], this last work developed independently from a different perspective.

C. Motion Planning Under Uncertainty

There is a vast literature on motion planning under uncertainty in robotics, excellent reviews of which may be found in texts such as [14], [36], [37] and examples of which range from early work on preimage backchaining [38] to very recent work on needle-steering using the stochastic motion roadmap [39]. As one example, we have drawn particular inspiration from work on sensorless manipulation [9]. In this work, like our own, the basic idea is to explicitly maintain the set of all possible robot configurations and to select a sequence of actions that reduces the size of this set and drives it toward some goal configuration. Carefully selected primitive operations can make this easier. For example, sensorless manipulation strategies often use a sequential composition of primitive operations, "squeezing" a part either virtually with a programmable force field or simply between two flat, parallel plates [40]. Sensorless manipulation strategies also may take advantage of limit cycle behavior, for example engineering fixed points and basins of attraction so that parts only exit a feeder when they reach the correct orientation [41], [42]. These two strategies have been applied to a much wider array of mechanisms such as vibratory bowls and tables [43], [44] or assembly lines [40], [45], [46], and have also been extended to situations with stochastic uncertainty [47], [48] and closed-loop feedback [49], [50]. Our interest in this particular collection of work also stems from our belief that ensemble control theory may provide new insight into sensorless manipulation of many objects at once.

III. Analysis of Controllability

In this section, we will establish controllability results for the system (3). Our method of approach will closely follow the one taken in [1]–[8]. First, we state formally what it means to be ensemble controllable.

Definition 1: Consider the family of control systems

$$\dot{q}(t, \epsilon) = f\left(q(t, \epsilon), u(t), t, \epsilon\right),$$

where $q \in \mathcal{Q} \subset \mathbb{R}^n$, $u \in \mathcal{U} \subset \mathbb{R}^m$, $\epsilon \in [1-\delta, 1+\delta]$ for $0 \leq \delta < 1$, and f is a smooth function. This family is *ensemble controllable* on the function space $L_2\left([1-\delta, 1+\delta], \mathcal{Q}\right)$ if for all $\mu > 0$ and continuous $q_{\text{start}}, q_{\text{goal}} \in L_2\left([1-\delta, 1+\delta], \mathcal{Q}\right)$ there exists $T > 0$ and piecewise-continuous $u \colon [0, T] \to \mathcal{U}$ such that $q(0, \epsilon) = q_{\text{start}}(\epsilon)$ and $\|q(T, \epsilon) - q_{\text{goal}}(\epsilon)\| \leq \mu$ for all $\epsilon \in [1-\delta, 1+\delta]$.

Note that this definition is a slight generalization of what appeared in Section I, since we allow q_{start} and q_{goal} to be arbitrary functions of ϵ. As pointed out by [31], the reader should interpret being ensemble controllable as being approximately controllable on $L_2\left([1-\delta, 1+\delta], \mathcal{Q}\right)$.

A. Finding a Controllable Subsystem

We begin with a proof by construction of the following negative result, which was originally suggested by [3].

Theorem 2: If $\delta > 0$, then the system (3) is not ensemble controllable.

Proof: Notice that for any u_1 and u_2, we have

$$\dot{\theta}(t, \epsilon) = \epsilon u_2(t).$$

As a consequence, if we define an auxiliary state $\gamma(t)$ such that $\gamma(0) = 0$ and

$$\dot{\gamma}(t) = u_2(t),$$

then it is clear that

$$\theta(t, \epsilon) = \theta(0, \epsilon) + \epsilon\gamma(t)$$

for all $\epsilon \in [1-\delta, 1+\delta]$, where we assume without loss of generality that we are working in the coordinates of a single local chart on S^1. For any $\Delta\theta > 0$, choose

$$q_{\text{start}}(\epsilon) = \left(x_{\text{start}}(\epsilon), y_{\text{start}}(\epsilon), \theta_{\text{start}}(\epsilon)\right)$$

and

$$q_{\text{goal}}(\epsilon) = (x_{\text{goal}}(\epsilon), y_{\text{goal}}(\epsilon), \theta_{\text{goal}}(\epsilon))$$

satisfying

$$\theta_{\text{goal}}(\epsilon) - \theta_{\text{start}}(\epsilon) = \Delta\theta$$

for all $\epsilon \in [1-\delta, 1+\delta]$, and let $\mu = \delta\Delta\theta/2$. We have

$$\begin{aligned}\|q(T,\epsilon) - q_{\text{goal}}(\epsilon)\| &\geq \|\theta(T,\epsilon) - \theta_{\text{goal}}(\epsilon)\| \\ &= \|\theta(0,\epsilon) + \epsilon\gamma(T) - \theta_{\text{goal}}(\epsilon)\| \\ &= \|\epsilon\gamma(T) - (\theta_{\text{goal}}(\epsilon) - \theta_{\text{start}}(\epsilon))\| \\ &= \|\epsilon\gamma(T) - \Delta\theta\|.\end{aligned}$$

Since we have assumed $\delta > 0$, then for any $\gamma(T)$ there exists some $\epsilon \in [1-\delta, 1+\delta]$ at which

$$\|\epsilon\gamma(T) - \Delta\theta\| > \mu,$$

and so (3) is not ensemble controllable by definition. ■

This result suggests the construction of a subsystem that, as we will show in the following section, is ensemble controllable. We write the configuration of this subsystem as

$$p(t,\epsilon) = (x(t,\epsilon), y(t,\epsilon), \gamma(t)),$$

where $\gamma(t)$ is the auxiliary state we introduced in the proof of Theorem 2. We have just shown that the evolution of this subsystem is governed by the alternate kinematic model

$$\dot{p}(t,\epsilon) = \epsilon h_1(p(t,\epsilon),\epsilon)\, u_1(t) + h_2(p(t,\epsilon),\epsilon)\, u_2(t), \tag{4}$$

where

$$\begin{aligned} h_1(p(t,\epsilon),\epsilon) &= \begin{bmatrix}\cos(\theta(0,\epsilon)+\epsilon\gamma(t)) \\ \sin(\theta(0,\epsilon)+\epsilon\gamma(t)) \\ 0\end{bmatrix} \\ h_2(p(t,\epsilon),\epsilon) &= \begin{bmatrix}0\\0\\1\end{bmatrix}\end{aligned} \tag{5}$$

and $\theta(0,\epsilon)$ is the initial heading given by q_{start}, as before. For convenience, we will abbreviate

$$\begin{aligned} c(t,\epsilon) &= \cos(\theta(0,\epsilon)+\epsilon\gamma(t)) \\ s(t,\epsilon) &= \sin(\theta(0,\epsilon)+\epsilon\gamma(t))\end{aligned}$$

so that

$$h_1(p(t,\epsilon),\epsilon) = \begin{bmatrix} c(t,\epsilon) \\ s(t,\epsilon) \\ 0\end{bmatrix}. \tag{6}$$

Since there is no longer any functional dependence of $p_3(t,\epsilon)$ on ϵ, it is clear that we have removed the feature of (3) that allowed us to conclude a lack of controllability. We will see that the resulting subsystem (4) is, in fact, controllable.

Before proceeding, notice that the vector field h_1 in (5) may be expressed

$$h_1(p(t,\epsilon),\epsilon) = R(\epsilon)\begin{bmatrix}\cos(\epsilon\gamma(t)) \\ \sin(\epsilon\gamma(t)) \\ 0\end{bmatrix}$$

where

$$R(\epsilon) = \begin{bmatrix}\cos\theta(0,\epsilon) & -\sin\theta(0,\epsilon) & 0 \\ \sin\theta(0,\epsilon) & \cos\theta(0,\epsilon) & 0 \\ 0 & 0 & 1\end{bmatrix},$$

so if we apply the transformation

$$p'(t,\epsilon) = R(\epsilon)^T p(t,\epsilon),$$

then without loss of generality it is always possible to assume that $\theta(0,\epsilon) = 0$ for all ϵ.

B. *Controllability By Polynomial Approximation*

We will now prove that the reduced subsystem derived in the previous section is ensemble controllable. We will do this by using repeated bracketing to get higher-order powers of ϵ, and then by using polynomial approximation to construct arbitrary vector flows. This approach is similar to what appears in [7], and involves computations that are reminiscent of [11].

Theorem 3: The system (4) is ensemble controllable.

Proof: Taking Lie brackets, we have

$$\begin{aligned}[\epsilon h_1, h_2] &= \epsilon\left(\frac{\partial h_2}{\partial p}h_1 - \frac{\partial h_1}{\partial p}h_2\right) \\ &= 0 - \epsilon\begin{bmatrix}0 & 0 & -\epsilon s \\ 0 & 0 & \epsilon c \\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix} \\ &= \epsilon^2\begin{bmatrix}s\\-c\\0\end{bmatrix}\end{aligned}$$

and

$$\begin{aligned}[[\epsilon h_1, h_2], h_2] &= 0 - \epsilon^2\begin{bmatrix}0 & 0 & \epsilon c \\ 0 & 0 & \epsilon s \\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}0\\0\\1\end{bmatrix} \\ &= -\epsilon^3\begin{bmatrix}c\\s\\0\end{bmatrix} \\ &= -\epsilon^3 h_1.\end{aligned}$$

Let us define

$$h_3 = \begin{bmatrix}-s\\c\\0\end{bmatrix},$$

so that $[\epsilon h_1, h_2] = -\epsilon^2 h_3$. Repeating this process, we can produce control vector fields of the form $\epsilon^{2i+1}h_1$ and $\epsilon^{2i+2}h_3$ for any $i \geq 0$. Since we have assumed that $\mathcal{U}$ is symmetric and that the affine hull of $\mathcal{U}$ is $\mathbb{R}^2$, then with piecewise-constant inputs (i.e., a sufficient number of "back-and-forth" manuevers) we can produce flows of the form

$$\exp(a_0\epsilon h_1)\cdots\exp\left(a_{k-1}\epsilon^{2k-1}h_1\right) = \exp\left(\sum_{i=0}^{k-1} a_i\epsilon^{2i+1}h_1\right)$$

and

$$\exp\left(b_0\epsilon^2 h_3\right)\cdots\exp\left(b_{k-1}\epsilon^{2k}h_3\right) = \exp\left(\sum_{i=0}^{k-1} b_i\epsilon^{2i+2}h_3\right)$$

for freely chosen coefficients $a, b \in \mathbb{R}^k$. Let

$$p_{\text{start}}(\epsilon) = (x_{\text{start}}(\epsilon), y_{\text{start}}(\epsilon), \gamma_{\text{start}})$$

and

$$p_{\text{goal}}(\epsilon) = (x_{\text{goal}}(\epsilon), y_{\text{goal}}(\epsilon), \gamma_{\text{goal}})$$

for any given continuous real-valued functions

$$x_{\text{start}}, y_{\text{start}}, x_{\text{goal}}, y_{\text{goal}} \in L_2\left([1-\delta, 1+\delta], \mathbb{R}\right)$$

and for any given $\gamma_{\text{start}}, \gamma_{\text{goal}} \in S^1$. Define

$$c = \gamma_{\text{goal}} - \gamma_{\text{start}}$$

and take

$$\begin{bmatrix}\alpha(\epsilon)\\ \beta(\epsilon)\end{bmatrix} = \begin{bmatrix}\cos c & \sin c\\ -\sin c & \cos c\end{bmatrix}\begin{bmatrix}x_{\text{goal}}(\epsilon) - x_{\text{start}}(\epsilon)\\ y_{\text{goal}}(\epsilon) - y_{\text{start}}(\epsilon)\end{bmatrix}$$

for all $\epsilon \in [1-\delta, 1+\delta]$, where continuity of α and β follows from continuity of $x_{\text{start}}, x_{\text{goal}}, y_{\text{start}}, y_{\text{goal}}$. We can represent the desired change in configuration by the flow

$$\exp\left(\beta(\epsilon)h_3\right)\exp\left(\alpha(\epsilon)h_1\right)\exp\left(ch_2\right).$$

The Stone-Weierstrass Theorem [51] tells us that given $\eta > 0$ and a continuous real function

$$\nu(\epsilon)\colon [1-\delta, 1+\delta] \to \mathbb{R},$$

there exists a polynomial function $\rho(\epsilon)$ such that

$$|\rho(\epsilon) - \nu(\epsilon)| < \eta$$

for all $\epsilon \in [\underline{\epsilon}, \overline{\epsilon}]$. An immediate corollary is that continuous real functions on the domain $[\underline{\epsilon}, \overline{\epsilon}] = [1-\delta, 1+\delta]$ for some $0 \leq \delta < 1$ can be uniformly approximated either by an odd polynomial or by an even polynomial. (This result would not be true on an arbitrary domain, which is why we restrict $\delta < 1$.) As a consequence, we can choose $a, b \in \mathbb{R}^k$ so that

$$\alpha(\epsilon) \approx \sum_{i=0}^{k-1} a_i \epsilon^{2i+1}$$
$$\beta(\epsilon) \approx \sum_{i=0}^{k-1} b_i \epsilon^{2i+2}$$

for $\epsilon \in [1-\delta, 1+\delta]$, with error vanishing in k. The time complexity of the resulting motion increases with k and with the number of switches required to approximate flows along each vector field $\epsilon^{2i+1}h_1$ and $\epsilon^{2i+2}h_3$, but remains finite for any given $\mu > 0$. Our result follows. ■

IV. Approximate Steering Algorithm

In the previous section, we showed that the subsystem (4) is ensemble controllable. Based on this result, we will now derive an approximate steering algorithm for this subsystem. Although the boundary conditions given to this algorithm could in general be arbitrary continuous functions $p_{\text{start}}(\epsilon)$ and $p_{\text{goal}}(\epsilon)$, for our application of interest—where (4) captures the range of possible outcomes for a single unicycle—these functions are always constant and have the form

$$p_{\text{start}}(\epsilon) = (x_{\text{start}}, y_{\text{start}}, \gamma_{\text{start}})$$
$$p_{\text{goal}}(\epsilon) = (x_{\text{goal}}, y_{\text{goal}}, \gamma_{\text{goal}}),$$

where we may as well assume that $\gamma_{\text{start}} = \gamma_{\text{goal}} = \gamma$. If we apply the transformation

$$\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \begin{bmatrix}\cos\gamma & \sin\gamma\\ -\sin\gamma & \cos\gamma\end{bmatrix}\begin{bmatrix}x_{\text{goal}} - x_{\text{start}}\\ y_{\text{goal}} - y_{\text{start}}\end{bmatrix},$$

then without loss of generality we may further assume that

$$p_{\text{start}}(\epsilon) = (0, 0, 0)$$
$$p_{\text{goal}}(\epsilon) = (\Delta x, \Delta y, 0).$$

Finally, we assume that $(1, 0) \in \mathcal{U}$ and that $(v, 1) \in \mathcal{U}$ for some $v \geq 0$, hence also that $(-1, 0), (-v, -1) \in \mathcal{U}$. We make this assumption primarily for convenience. Scaling either input would require only scaling the corresponding time for which it is applied, and taking the reflection $-\Delta y$ would directly address the case where $(v, -1) \in \mathcal{U}$. However, the fact that it is possible to "go straight" is important for the simplicity of our algorithm. If $(1, 0) \notin \mathcal{U}$, then we will assume that the corresponding control vector field is approximated by rapid switching, which is possible because the affine hull of $\mathcal{U}$ is $\mathbb{R}^2$. In any case, our model applies unchanged to both a differential-drive robot ($v = 0$) and a car-like robot ($v \neq 0$).

A. One Motion Primitive with Piecewise-Constant Inputs

Consider the following input for $\psi \geq 0$ and $a', b' \in \mathbb{R}$:

$$u(t) = \begin{cases}(v, 1) & 0 \leq t < \psi\\ (\operatorname{sgn} a', 0) & \psi \leq t < \psi + |a'|\\ (-v, -1) & \psi + |a'| \leq t < 3\psi + |a'|\\ (\operatorname{sgn} b', 0) & 3\psi + |a'| \leq t < 3\psi + |a'| + |b'|\\ (v, 1) & 3\psi + |a'| + |b'| \leq t < 4\psi + |a'| + |b'|.\end{cases}$$

We call this input a *motion primitive*. If $\gamma(0) = 0$, then the result of applying this motion primitive is to achieve

$$p(\Delta t, \epsilon) - p(0, \epsilon) = \begin{bmatrix}(a' + b')\epsilon\cos(\epsilon\psi)\\ (a' - b')\epsilon\sin(\epsilon\psi)\\ 0\end{bmatrix}$$

in time

$$\Delta t = 4\psi + |a'| + |b'|.$$

With the input transformation

$$a' = \frac{a+b}{2} \qquad\qquad b' = \frac{a-b}{2}$$

for freely chosen $a, b \in \mathbb{R}$, we can write this expression as

$$p(\Delta t, \epsilon) - p(0, \epsilon) = \begin{bmatrix}a\epsilon\cos(\epsilon\psi)\\ b\epsilon\sin(\epsilon\psi)\\ 0\end{bmatrix}$$

We will denote this motion primitive by the triple (a, b, ψ) and use it as the basis for our approximate steering algorithm.

B. Composition of Two Motion Primitives

Because our motion primitives leave γ invariant, we are free to concatenate them. For example, consider the sequential application of two primitives (a_1, b_1, ψ_1) and (a_2, b_2, ψ_2). If $\gamma(0) = 0$, then the result is to achieve

$$p(\Delta t, \epsilon) - p(0, \epsilon) = \begin{bmatrix}a_1\epsilon\cos(\epsilon\psi_1) + a_2\epsilon\cos(\epsilon\psi_2)\\ b_1\epsilon\sin(\epsilon\psi_1) + b_2\epsilon\sin(\epsilon\psi_2)\\ 0\end{bmatrix}$$

in time

$$\Delta t = (4\psi_1 + |a_1'| + |b_1'|) + (4\psi_2 + |a_2'| + |b_2'|),$$

where
$$a_i' = \frac{a_i + b_i}{2} \qquad b_i' = \frac{a_i - b_i}{2}$$
for $i \in \{1,2\}$. In fact, we can compose these two primitives in a slightly different way that achieves the same result in less time. Assume that $\psi_2 > \psi_1$. Consider the following input:
$$u(t) = \begin{cases} (v,1) & 0 \le t < \psi_1 \\ (\operatorname{sgn} a_1', 0) & \cdots \le t < \cdots + |a_1'| \\ (v,1) & \cdots \le t < \cdots + (\psi_2 - \psi_1) \\ (\operatorname{sgn} a_2', 0) & \cdots \le t < \cdots + |a_2'| \\ (-v,-1) & \cdots \le t < \cdots + (\psi_2 + \psi_1) \\ (\operatorname{sgn} b_1', 0) & \cdots \le t < \cdots + |b_1'| \\ (-v,-1) & \cdots \le t < \cdots + (\psi_2 - \psi_1) \\ (\operatorname{sgn} b_2', 0) & \cdots \le t < \cdots + |b_2'| \\ (v,1) & \cdots \le t < \cdots + \psi_2. \end{cases}$$
It is easy to verify that $p(\Delta t, \epsilon) - p(0, \epsilon)$ remains the same but that
$$\Delta t = (|a_1'| + |b_1'|) + (4\psi_2 + |a_2'| + |b_2'|),$$
which is lower than before by $4\psi_1$. Figure 1 shows an example, for which $\psi_1 = \pi/4$, $\psi_2 = \pi/2$, and $v = 1/2$.

C. Composition of Many Motion Primitives

We generalize our result of the previous section as follows. Given $\phi > 0$, consider a sequence of $k+1$ motion primitives
$$(a_{j+1}, b_j, \psi_j = j\phi)$$
for $j \in \{0, \ldots, k\}$, where we restrict $a_{k+1} = b_0 = 0$. We have indexed these primitives so that they are defined by the choice of $a, b \in \mathbb{R}^k$, where $a = (a_1, \ldots, a_k)$ and $b = (b_1, \ldots, b_k)$ as usual. We compose these primitives as in Section IV-B, noting that because $\psi_0 = 0$, the resulting inputs begin with translation and not with rotation. In particular, we have
$$u(t) = \begin{cases} (\operatorname{sgn} a_1', 0) & 0 \le t < |a_1'| \\ (v,1) & \cdots \le t < \cdots + \phi \\ (\operatorname{sgn} a_1', 0) & \cdots \le t < \cdots + |a_2'| \\ \quad\vdots & \\ (v,1) & \cdots \le t < \cdots + \phi \\ (\operatorname{sgn} a_k', 0) & \cdots \le t < \cdots + |a_k'| \\ (-v,-1) & \cdots \le t < \cdots + k\phi \\ (\operatorname{sgn} b_1', 0) & \cdots \le t < \cdots + |b_1'| \\ (-v,-1) & \cdots \le t < \cdots + \phi \\ (\operatorname{sgn} b_2', 0) & \cdots \le t < \cdots + |b_2'| \\ \quad\vdots & \\ (-v,-1) & \cdots \le t < \cdots + \phi \\ (\operatorname{sgn} b_k', 0) & \cdots \le t < \cdots + |b_k'| \\ (v,1) & \cdots \le t < \cdots + (k-1)\phi. \end{cases} \tag{7}$$
where
$$a' = \frac{1}{2}\left(\begin{bmatrix} a \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ b \end{bmatrix}\right) \qquad b' = \frac{1}{2}\left(\begin{bmatrix} a \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ b \end{bmatrix}\right).$$

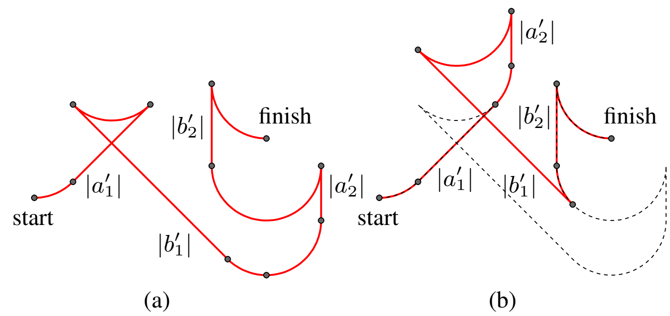

Fig. 1. Two different ways to compose motion primitives $(a_1, b_1, \psi_1 = \pi/4)$ and $(a_2, b_2, \psi_2 = \pi/2)$, where $v = 1/2$. In (a), they are concatenated. In (b), they are interwoven, with the same result but lower execution time. The concatenation is shown in (b) as a dotted line for comparison. In this figure, we are only showing the nominal path, corresponding to $\epsilon = 1$.

As before, it is easy to verify that
$$p(\Delta t, \epsilon) - p(0, \epsilon) = \begin{bmatrix} \sum_{j=1}^k a_j \epsilon \cos(\epsilon(j-1)\phi) \\ \sum_{j=1}^k b_j \epsilon \sin(\epsilon j \phi) \\ 0 \end{bmatrix},$$
where
$$\Delta t = 4(k-1)\phi + \sum_{i=1}^k (|a_i'| + |b_i'|). \tag{8}$$
As in Section III, our problem has been reduced to function approximation. Given $\mu > 0$ and $(\Delta x, \Delta y) \in \mathbb{R}^2$, we need to find $\phi > 0$ and $a, b \in \mathbb{R}^k$ for sufficiently large k so that
$$\left| \Delta x - \sum_{j=1}^k a_j \epsilon \cos(\epsilon(j-1)\phi) \right| \le \mu$$
and
$$\left| \Delta y - \sum_{j=1}^k b_j \epsilon \sin(\epsilon j \phi) \right| \le \mu$$
for all $\epsilon \in [1-\delta, 1+\delta]$. The resulting input (7) would then be a solution to the approximate steering problem for (4).

D. Achieving Error of a Particular Order

We can express the result
$$\Delta p_1(\epsilon) = p_1(\Delta t, \epsilon) - p_1(0, \epsilon)$$
$$\Delta p_2(\epsilon) = p_2(\Delta t, \epsilon) - p_2(0, \epsilon)$$
of applying (7) as Taylor series about $\epsilon = 1$:
$$\Delta p_1(\epsilon) = \Delta p_1(1) + \left(\left.\frac{\partial \Delta p_1}{\partial \epsilon}\right|_{\epsilon=1} \right)(\epsilon - 1) + \cdots$$
$$\Delta p_2(\epsilon) = \Delta p_2(1) + \left(\left.\frac{\partial \Delta p_2}{\partial \epsilon}\right|_{\epsilon=1} \right)(\epsilon - 1) + \cdots.$$
Each series has the form
$$\Delta p_1(\epsilon) = \sum_{i=1}^k r_i (\epsilon - 1)^{i-1} + O\left(|\epsilon - 1|^k\right)$$
$$\Delta p_2(\epsilon) = \sum_{i=1}^k s_i (\epsilon - 1)^{i-1} + O\left(|\epsilon - 1|^k\right),$$

where we collect $r = (r_1, \dots, r_k)$ and $s = (s_1, \dots, s_k)$ so that $r, s \in \mathbb{R}^k$. Explicit formulas for r and s are given by

$$\begin{aligned} r &= Aa \\ s &= Bb, \end{aligned} \tag{9}$$

where the matrices $A, B \in \mathbb{R}^{k \times k}$ have elements

$$A_{ij} = \frac{1}{(i-1)!} \left(\frac{\partial^{i-1} \left(\epsilon \cos \left(\epsilon (j-1) \phi \right) \right)}{\partial \epsilon^{i-1}} \bigg|_{\epsilon=1} \right)$$
$$B_{ij} = \frac{1}{(i-1)!} \left(\frac{\partial^{i-1} \left(\epsilon \sin \left(\epsilon j \phi \right) \right)}{\partial \epsilon^{i-1}} \bigg|_{\epsilon=1} \right)$$

for all $i, j \in \{1, \dots, k\}$. Note that A and B do not depend on ϵ. To approximate $\Delta x = 1$ and $\Delta y = 1$ with error that is of order k in $|\epsilon - 1|$, we require only a solution a, b to the system of linear equations (9) that results in

$$\begin{aligned} r &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T \\ s &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}^T. \end{aligned}$$

Both A and B are square matrices, so assuming both are non-singular (and well-conditioned)—which will hold for almost all choices of the angle ϕ—then (9) has a unique solution. An immediate consequence is that achievable error decreases exponentially in the number $k+1$ of primitives, a result that was shown empirically in [25].

By linearity, if the parameters a and b achieve

$$(\Delta p_1, \Delta p_2) = (1, 1),$$

then the scaled parameters $a\Delta x$ and $b\Delta y$ achieve

$$(\Delta p_1, \Delta p_2) = (\Delta x, \Delta y)$$

for arbitrary Δx and Δy. In other words, scaling a single, precomputed maneuver gets you everywhere for free. Subsequently, we need only compute

$$a' = \frac{1}{2} \left(\begin{bmatrix} a \\ 0 \end{bmatrix} \Delta x + \begin{bmatrix} 0 \\ b \end{bmatrix} \Delta y \right)$$
$$b' = \frac{1}{2} \left(\begin{bmatrix} a \\ 0 \end{bmatrix} \Delta x - \begin{bmatrix} 0 \\ b \end{bmatrix} \Delta y \right).$$

One advantage of this strategy over the one used in [25] is that we no longer have to sample ϵ in order to compute the parameters a and b. Doing so had previously introduced approximation error that was difficult to quantify. Now, the series expansion gives us an explicit bound on this error. In particular, to achieve a tolerance $\mu > 0$, we simply choose any integer $k > 0$ that satisfies $\delta^{k-1} < \mu$.

We can also quantify the trade-off between the time cost Δt and the resulting uncertainty. The total elapsed time to reach $(\Delta x, \Delta y)$ with kth-order error in $|\epsilon - 1|$ is given by (8). Direct computation verifies that $|a'_i|$ and $|b'_i|$ decay rapidly with i, so the term $4(k-1)\phi$ dominates the elapsed time. As a consequence, the time cost is $O(k)$. Note that if switching is required to generate $(u_1, u_2) = (1, 0)$—i.e., if $(1, 0) \notin \mathcal{U}$—this will only scale the cost by a constant factor.

Finally, we consider the total distance traveled in the workspace. If $\epsilon = 1$, this distance is given by

$$d(\Delta x, \Delta y) = 4(k-1)\phi v + \sum_{i=1}^{k} \left(|a'_i| + |b'_i| \right).$$

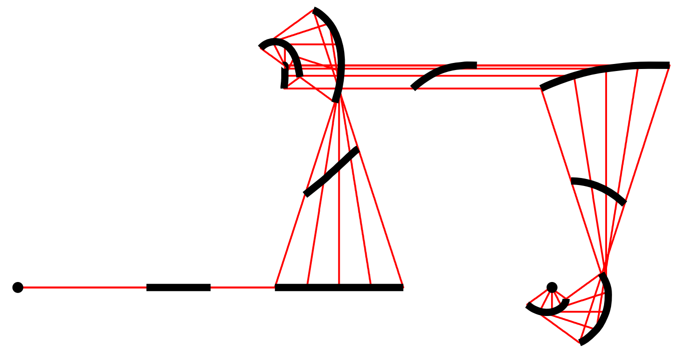

Fig. 2. An ensemble with $\epsilon \in [0.8, 1.2]$ moving a unit distance in the x direction achieving 4th-order error in $|\epsilon - 1|$, which corresponds to a maximum error bound of $0.2^3 = 0.008$. Thin red lines show the path followed for particular values of ϵ. The actual robot follows only one of these paths. Thick black lines show the entire ensemble at instants of time.

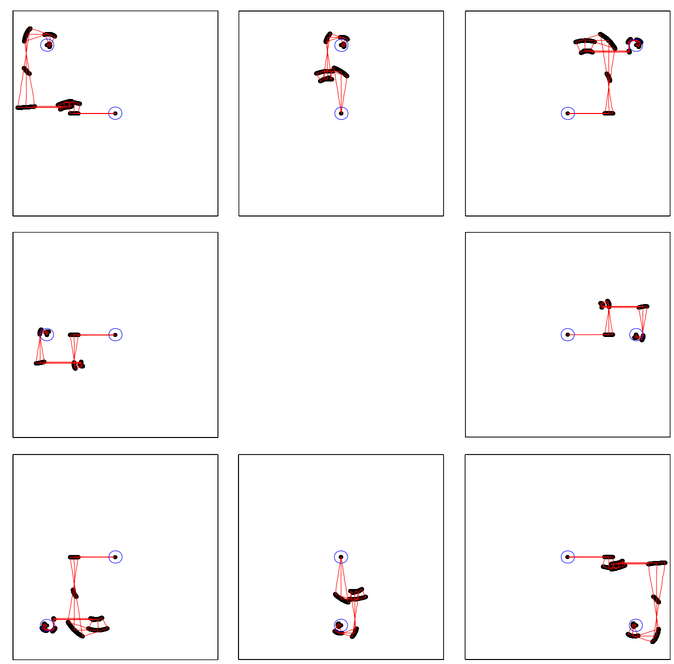

Fig. 3. Example trajectories for $k = 4$ and $\phi = \pi/2$. All of them are scaled versions of the maneuver shown in Fig. 2. Thin red lines are particular values of ϵ, thick black lines are the entire ensemble at instants of time.

We may compute an upper bound on d by solving the following convex optimization problem, which is linear in the decision variables Δx and Δy:

$$\begin{aligned} \text{minimize} \quad & d(\Delta x, \Delta y) \\ \text{subject to} \quad & |\Delta x| \leq 1 \\ & |\Delta y| \leq 1. \end{aligned}$$

Call the solution to this problem d_{min}. Recall that $\epsilon \in [1-\delta, 1+\delta]$ and $0 \leq \delta < 1$, so the distance traveled for any ϵ is at most $2d_{\text{min}}$. As a corollary, we know that $(x(t,\epsilon), y(t,\epsilon))$ remains always inside a ball of radius $2d_{\text{min}}$ during the application of our steering algorithm. This interesting result indicates that it might be possible to prove some form of small-time local controllability as the basis for extending our work from steering to motion planning (e.g., as in [52], [53]), although

COMPUTEPRIMITIVE(ϕ, k)

Given an angle ϕ and a non-negative integer k, do the following:

- Compute the elements of $A, B \in \mathbb{R}^{k\times k}$ according to

$$A_{ij} = \frac{1}{(i-1)!}\left(\frac{\partial^{i-1}\left(\epsilon\cos\left(\epsilon(j-1)\phi\right)\right)}{\partial\epsilon^{i-1}}\bigg|_{\epsilon=1}\right)$$
$$B_{ij} = \frac{1}{(i-1)!}\left(\frac{\partial^{i-1}\left(\epsilon\sin\left(\epsilon j\phi\right)\right)}{\partial\epsilon^{i-1}}\bigg|_{\epsilon=1}\right)$$

  for all $i, j \in \{1, \ldots, k\}$.
- Compute $a, b \in \mathbb{R}^k$ according to

$$a = A^{-1}r$$
$$b = B^{-1}s,$$

  where

$$r = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}^T$$
$$s = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}^T.$$

Return (a, b).

Fig. 4. The algorithm that we use to precompute a motion primitive.

COMPUTEINPUT$(\phi, k, x_{\text{start}}, y_{\text{start}}, x_{\text{goal}}, y_{\text{goal}}, \gamma)$

Given an angle ϕ, a non-negative integer k, a start configuration $(x_{\text{start}}, y_{\text{start}}, \gamma)$, and a goal configuration $(x_{\text{goal}}, y_{\text{goal}}, \gamma)$, do the following:

- Compute the motion primitive

$$(a, b) = \text{COMPUTEPRIMITIVE}(\phi, k).$$

- Compute the desired change in Cartesian position

$$\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \begin{bmatrix}\cos\gamma & \sin\gamma\\ -\sin\gamma & \cos\gamma\end{bmatrix}\begin{bmatrix}x_{\text{goal}} - x_{\text{start}}\\ y_{\text{goal}} - y_{\text{start}}\end{bmatrix}.$$

- Compute the parameters

$$a' = \frac{1}{2}\left(\begin{bmatrix}a\\0\end{bmatrix}\Delta x + \begin{bmatrix}0\\b\end{bmatrix}\Delta y\right)$$
$$b' = \frac{1}{2}\left(\begin{bmatrix}a\\0\end{bmatrix}\Delta x - \begin{bmatrix}0\\b\end{bmatrix}\Delta y\right).$$

- Compute the piecewise-constant input

$$u(t) = \begin{cases}(\operatorname{sgn} a_1', 0) & 0 \le t < |a_1'|\\ (v, 1) & \cdots \le t < \cdots + \phi\\ (\operatorname{sgn} a_1', 0) & \cdots \le t < \cdots + |a_2'|\\ \quad\vdots & \\ (v, 1) & \cdots \le t < \cdots + \phi\\ (\operatorname{sgn} a_k', 0) & \cdots \le t < \cdots + |a_k'|\\ (-v, -1) & \cdots \le t < \cdots + k\phi\\ (\operatorname{sgn} b_1', 0) & \cdots \le t < \cdots + |b_1'|\\ (-v, -1) & \cdots \le t < \cdots + \phi\\ (\operatorname{sgn} b_2', 0) & \cdots \le t < \cdots + |b_2'|\\ \quad\vdots & \\ (-v, -1) & \cdots \le t < \cdots + \phi\\ (\operatorname{sgn} b_k', 0) & \cdots \le t < \cdots + |b_k'|\\ (v, 1) & \cdots \le t < \cdots + (k-1)\phi.\end{cases}$$

Return u.

Fig. 5. Our approximate steering algorithm. It acts only to scale the primitive generated by the subroutine COMPUTEPRIMITIVE, which need only be called once for given ϕ and k. We recommend choosing $\phi = \pi/2$ and the smallest integer k such that $\delta^{k-1} < \mu$ for a given tolerance $\mu > 0$.

it is not obvious yet how to proceed.

Figures 4-5 provide pseudo-code that implements our approximate steering algorithm. We emphasize that this algorithm produces an open-loop input trajectory that neither requires nor takes advantage of sensor feedback.

E. Results in Simulation

Figure 2 shows the results of applying our approximate steering algorithm to an ensemble control system of the form (4) for which $\delta = 0.2$ and $(\Delta x, \Delta y) = (1, 0)$. In this example, we chose $k = 4$, so that maximum error is expected to be $O(\delta^4)$. Equivalently, we expect that

$$0.008 = \delta^3 > \max_{\epsilon\in[1-\delta,1+\delta]}\{\Delta p_1(\epsilon) - \Delta x\}$$
$$0.008 = \delta^3 > \max_{\epsilon\in[1-\delta,1+\delta]}\{\Delta p_2(\epsilon) - \Delta y\}.$$

We chose $\phi = \pi/2$. As a consequence, the matrices A and B have a very simple form:

$$A = \begin{bmatrix}1 & 0 & -1 & 0\\ 1 & -\pi/2 & -1 & 3\pi/2\\ 0 & -\pi/2 & \pi^2/2 & 3\pi/2\\ 0 & \pi^3/48 & \pi^2/16 & -9\pi^3/16\end{bmatrix}$$

and

$$B = \begin{bmatrix}1 & 0 & -1 & 0\\ 1 & -\pi & -1 & 2\pi\\ -\pi^2/8 & -\pi & 9\pi^2/8 & 2\pi\\ -\pi^2/8 & \pi^3/6 & 9\pi^2/8 & -4\pi^3/3\end{bmatrix}.$$

The linear equations (9) can be solved exactly to produce

$$a = \begin{bmatrix}1 + (2/\pi^2)\\ 3\left(8 + 3\pi^2\right)/\left(4\pi^3\right)\\ 2/\pi^2\\ \left(24 + \pi^2\right)/\left(12\pi^3\right)\end{bmatrix} \quad\text{and}\quad b = \begin{bmatrix}(9/8) + \left(1/\pi^2\right)\\ \left(6 + 4\pi^2\right)/\left(3\pi^3\right)\\ (1/8) + \left(1/\pi^2\right)\\ \left(6 + \pi^2\right)/\left(6\pi^3\right)\end{bmatrix}.$$

We also compute exactly the bound on total distance traveled, in this case achieved when $(\Delta x, \Delta y) = (-1, -1)$:

$$d_{\text{min}} = \frac{9}{4} + \frac{6 + \pi(8 + 3\pi)}{2\pi^3} \approx 3.23.$$

We verify in simulation that the maximum error is 0.003 and that the distance traveled is 2.41, both satisfying our predicted bounds. Figure 3 shows the same precomputed motion primitive scaled to reach a variety of goal configurations.

V. HARDWARE EXPERIMENTS

In this section we apply our approximate steering algorithm to a differential-drive robot with unknown but bounded wheel radius. First, we describe the robot that we used. Then, we show that (2) is an appropriate model of this robot. Finally, we show the results of hardware experiments.

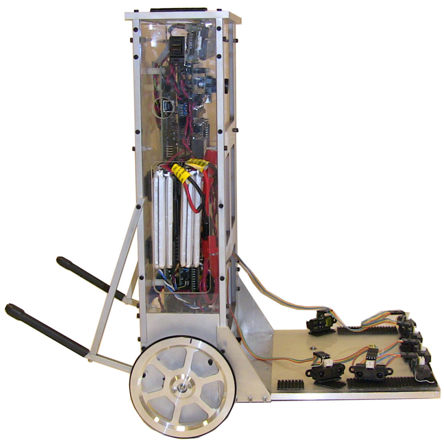

Fig. 6. The differential-drive robot used for experimental validation.

Fig. 7. Four wheel sizes used for experimental validation. These wheels are 10.16, 10.48, 12.7 and 15.24 cm in diameter.

A. Experimental Setup

Figure 6 shows the robot we used in our experiments. It is a differential-drive robot with a caster wheel in front for stability. It moves on a flat tile floor and uses only dead-reckoning for navigation. In particular, the robot runs a feedback control loop to read the wheel encoders, update a dead-reckoning position estimate, and regulate the speed of each motor. Although we use no other sensors for feedback control, global position data is available from an off-board vision system for later analysis. This vision system records pose information at 27Hz with a position accuracy of 2cm and an orientation accuracy of $1°$.

Before conducting our experiments, we applied a standard calibration procedure to find the effective wheelbase and wheel radius in order to reduce systematic dead-reckoning error [16]. The calibration was done with wheels of diameter 12.7cm. However, these wheels are interchangeable—in our experiments, we used four sets that varied between 10.16-15.24cm in diameter, as shown in Fig. 7. We did not recalibrate for these other wheels, and assumed that the wheel diameter was unknown but bounded in the set $[10.2, 15.2]$, or in other words the set $[0.8, 1.2]$ relative to the nominal diameter 12.7cm.

B. Application of the Model to a Differential-Drive Robot

We will show that

$$\dot{q}(t) = \epsilon \left(g_1(q(t))u_1(t) + g_2(q(t))u_2(t) \right)$$

is a valid kinematic model of our robot, where

$$g_1(q) = \begin{bmatrix} \cos q_3 \\ \sin q_3 \\ 0 \end{bmatrix} \qquad \text{and} \qquad g_2(q) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

It suffices to show that the forward speed v and turning rate ω of a differential-drive robot with unknown but bounded wheel radius are given by $v = \epsilon u_1$ and $\omega = \epsilon u_2$, respectively, for control inputs $u_1, u_2 \in \mathbb{R}$. Recall that for wheel radius r and wheel separation l, the forward speed and turning rate of a differential-drive robot are given by

$$v = \frac{r(\omega_R + \omega_L)}{2} \qquad \text{and} \qquad \omega = \frac{r(\omega_R - \omega_L)}{l},$$

where ω_R and ω_L are the angular velocities of the right and left wheels, respectively. Assume that the wheel radius, a positive constant, is unknown but bounded according to $r \in [r_{\text{min}}, r_{\text{max}}]$. If we define

$$\bar{r} = \frac{r_{\text{max}} + r_{\text{min}}}{2} \qquad \text{and} \qquad \delta = \frac{r_{\text{max}} - r_{\text{min}}}{2\bar{r}}$$

then we can write $r = \epsilon \bar{r}$ for some $\epsilon \in [1-\delta, 1+\delta]$, so that

$$v = \epsilon \left(\frac{\bar{r}(\omega_R + \omega_L)}{2} \right) \qquad \text{and} \qquad \omega = \epsilon \left(\frac{\bar{r}(\omega_R - \omega_L)}{l} \right).$$

This expression simplifies if we select wheel angular velocities

$$\omega_R = \frac{2u_1 + bu_2}{2\bar{r}} \qquad \text{and} \qquad \omega_L = \frac{2u_1 - bu_2}{2\bar{r}}$$

for any given $u_1, u_2 \in \mathbb{R}$, so that

$$v = \epsilon u_1 \qquad \text{and} \qquad \omega = \epsilon u_2,$$

and we have our result.

C. Experimental Results

Figure 8 shows the results of our experiments, which successfully validated our approach. The start configuration was $(0, 0, 0)$. The goal configuration was $(4.25\text{m}, 2.25\text{m}, 0)$. The value of k was chosen to achieve an error tolerance of 2cm. We applied the algorithm described in Section IV to generate a single input trajectory that was applied in open-loop. Five runs were recorded for each wheel size. All of the resulting trajectories reached a small neighborhood of the goal position, as shown in Fig. 9 and reported in aggregate in Table I. The size of this neighborhood is slightly larger than the predicted tolerance of 2cm. This error is due largely to drift as a result of wheel slip, gear backlash, surface irregularities, wheel flex, and other disturbances. Another contributing factor is that different wheels are made of different materials. The 10.48cm wheels are aluminum with rubber o-rings stretched over the rim, while the other wheels are ABS plastic with a molder rubber traction ring on the rim. The edge of each plastic wheel has a rectangular cross-section, making the effective wheel base slightly larger than for the aluminum wheels. The vision system also adds to observed error (although we emphasize that this vision system was used only for data collection and not for closed-loop feedback in our experiments). In particular, ground truth position information was calculated from fiducial markers on the top of the robot. These markers were level and centered over the wheelbase for the 10.48cm wheels, but tilted by $10°$ for the largest wheels.

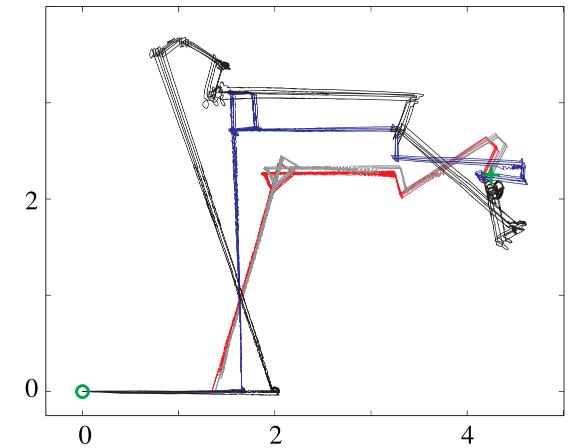

Fig. 8. Ground truth data gathered from the camera system. Five runs for each wheel set are shown. Loops at the corners are artifacts from the camera system. Red, grey, blue and black plots correspond to 10.16, 10.48, 12.7 and 15.24 cm wheels, respectively. Units are in meters.

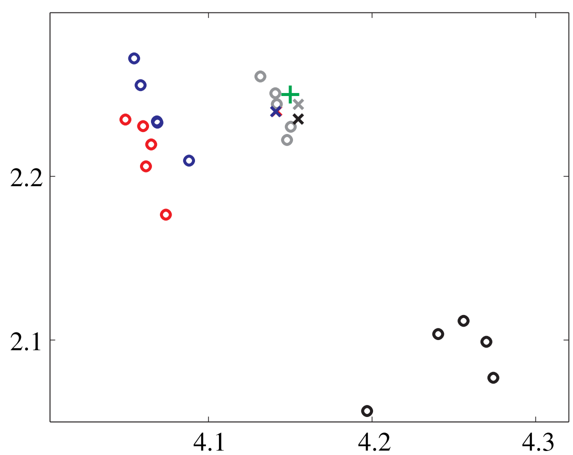

Fig. 9. Ending position for each run. Green '+' for goal position, 'x' for expected ending position under zero odometry drift, 'o' for actual ending positions. Red, grey, blue and black plots correspond to 10.16, 10.48, 12.7 and 15.24 cm wheels. Units are in meters (note the zoomed scale).

TABLE I
IN-GROUP ERROR MEASUREMENTS

wheel diam(cm)	distance mean(m)	distance var (m^2)	θ mean(rad)	θ var$(rad)^2$
10.16	0.10	3.2e-5	-0.002	1.9e-6
10.48	0.02	4.1e-5	-0.002	6.1e-7
12.70	0.08	1.2e-4	-0.008	2.5e-5
15.24	0.19	2.2e-4	-0.017	1.9e-4

VI. CONCLUSION

In this paper we applied the framework of ensemble control theory to derive an approximate steering algorithm that brings a nonholonomic unicycle to with an arbitrarily small neighborhood of any given Cartesian position despite model perturbation that scales both the forward speed and the turning rate by an unknown but bounded constant. This algorithm has trivial computational complexity, requiring only the solution of linear equations. We validated our approach using a differential-drive robot with unknown but bounded wheel radius and showed the results with hardware experiments. We hope that these results stimulate interest in ensemble control theory and provoke a new line of inquiry that may ultimately lead to practical application in robotics.

VII. ACKNOWLEDGEMENTS

The authors wish to thank Chad Burns, Dušan Stipanović and Dan Block for help with experiments, and Seth Hutchinson, Andy Ruina, and Mark Psiaki for helpful discussion. This material is based upon work supported by the National Science Foundation under Grant Nos. 0931871 and 0956362.

REFERENCES

[1] R. W. Brockett and N. Khaneja, "On the control of quantum ensembles," in *System Theory: Modeling, Analysis and Control*, T. Djaferis and I. Schick, Eds. Kluwer Academic Publishers, 1999.

[2] N. Khaneja, "Geometric control in classical and quantum systems," Ph.D. dissertation, Harvard University, 2000.

[3] J.-S. Li, "Control of inhomogeneous ensembles," Ph.D. dissertation, Harvard University, May 2006.

[4] J.-S. Li and N. Khaneja, "Control of inhomogeneous quantum ensembles," *Physical Review A (Atomic, Molecular, and Optical Physics)*, vol. 73, no. 3, p. 030302, 2006.

[5] ——, "Ensemble controllability of the bloch equations," in *IEEE Conf. Dec. Cont.*, San Diego, CA, Dec. 2006, pp. 2483–2487.

[6] ——, "Ensemble control of linear systems," in *IEEE Conf. Dec. Cont.*, New Orleans, LA, USA, Dec. 2007, pp. 3768–3773.

[7] ——, "Ensemble control of bloch equations," *IEEE Trans. Autom. Control*, vol. 54, no. 3, pp. 528–536, Mar. 2009.

[8] J.-S. Li, "Ensemble control of finite-dimensional time-varying linear systems," *IEEE Trans. Autom. Control*, vol. 56, no. 2, pp. 345–357, Feb 2011.

[9] M. A. Erdmann and M. T. Mason, "An exploration of sensorless manipulation," *IEEE J. Robot. Autom.*, vol. 4, no. 4, pp. 369–379, Aug. 1988.

[10] H. J. Sussmann and V. Jurdjevic, "Controllability of nonlinear systems," *Journal of Differential Equations*, vol. 12, no. 1, pp. 95 – 116, 1972. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0022039672900071

[11] G. Lafferriere and H. Sussmann, "A differential geometric approach to motion planning," in *Nonholonomic Motion Planning*, Z. Li and J. Canny, Eds. Kluwer, 1993, pp. 235–270.

[12] B. Siciliano, *Robotics: modelling, planning and control*. London: Springer, 2009.

[13] P. Lucibello and G. Oriolo, "Robust stabilization via iterative state steering with an application to chained-form systems," *Automatica*, vol. 37, no. 1, pp. 71–79, Jan. 2001.

[14] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[15] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*, 2nd ed., ser. Intelligent robotics and autonomous agents series. Cambridge, MA: MIT Press, 2011.

[16] J. Borenstein and L. Feng, "Umbmark: a benchmark test for measuring odometry errors in mobile robots," in *SPIE*, W. J. Wolfe and C. H. Kenyon, Eds., vol. 2591, no. 1, 1995, pp. 113–124.

[17] G. Antonelli and S. Chiaverini, "Linear estimation of the physical odometric parameters for differential-drive mobile robots," *Autonomous Robots*, vol. 23, no. 1, pp. 59–68, Jul. 2007.

[18] A. Martinelli, N. Tomatis, and R. Siegwart, "Simultaneous localization and odometry self calibration for mobile robot," *Autonomous Robots*, vol. 22, no. 1, pp. 75–85, Jan. 2007.

[19] R. W. Brockett and L. Dai, *Non-holonomic kinematics and the role of elliptic functions in constructive controllability*. Kluwer Academic Publishers, 1993, ch. 1.

[20] Z. Li and J. Canny, "Motion of two rigid bodies with rolling constraint," *IEEE Trans. Robot. Autom.*, vol. 6, no. 1, pp. 62–72, Feb. 1990.

[21] V. Jurdjevic, "The geometry of the plate-ball problem," *Archive For Rational Mechanics and Analysis*, vol. 124, no. 4, pp. 305–328, 1993.

[22] A. Marigo and A. Bicchi, "Rolling bodies with regular surface: controllability theory and applications," *IEEE Trans. Autom. Control*, vol. 45, no. 9, pp. 1586–1599, Sep. 2000.

[23] A. A. G. Requicha, "Toward a theory of geometric tolerancing," *International Journal of Robotics Research*, vol. 2, no. 4, pp. 45–60, 1983.

[24] S. Akella and M. T. Mason, "Orienting toleranced polygonal parts," *The International Journal of Robotics Research*, vol. 19, no. 12, pp. 1147–1170, 2000. [Online]. Available: http://ijr.sagepub.com/content/19/12/1147.abstract

[25] A. Becker and T. Bretl, "Motion planning under bounded uncertainty using ensemble control," in *Robotics: Science and Systems (RSS)*, 2010.

[26] R. Murray, Z. X. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.

[27] S. V. D. Meer, "Stochastic cooling and the accumulation of antiprotons," in *Nobel Lectures in Physics*, T. Frängsmyr, Ed. Singapore: World Scientific, 1993.

[28] N. Michael and V. Kumar, "Controlling shapes of ensembles of robots of finite size with nonholonomic constraints," in *Robotics: Science and Systems*, Zurich, Switzerland, 2008.

[29] L. U. Odhner and H. Asada, "Stochastic recruitment control of large ensemble systems with limited feedback," *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme*, vol. 132, no. 4, Jul. 2010.

[30] P. Rouchon, "Motion planning, equivalence, infinite dimensional systems," *Int. J. Appl. Math. Comput. Sci.*, vol. 11, no. 1, pp. 165–188, 2001.

[31] K. Beauchard, J.-M. Coron, and P. Rouchon, "Controllability issues for continuous-spectrum systems and ensemble controllability of bloch equations," *Communications in Mathematical Physics*, vol. 296, pp. 525–557, 2010.

[32] G. E. Dullerud and F. G. Paganini, *A course in robust control theory: a convex approach*. New York: Springer, 2000, vol. 36. [Online]. Available: http://www.loc.gov/catdir/enhancements/fy0816/99046358-d.html

[33] N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *Journal of Dynamic Systems, Measurement, and Control*, vol. 112, no. 1, pp. 76–82, 1990. [Online]. Available: http://link.aip.org/link/?JDS/112/76/1

[34] M. Fischer and M. L. Psiaki, "Robustified control design by an approximate min-max technique," in *AIAA Guidance, Navigation and Control Conference*, Monterey, CA, Aug. 1993, pp. 949–957.

[35] J.-S. Li, "A new perspective on control of uncertain complex systems," in *IEEE Conf. Dec. Cont.*, 2009, pp. 708 –713.

[36] M. T. Mason, *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[37] S. M. LaValle, *Planning algorithms*. New York, NY: Cambridge University Press, 2006.

[38] M. Erdmann, "Using backprojections for fine motion planning with uncertainty," *International Journal of Robotics Research*, vol. 5, no. 1, pp. 19–45, 1986.

[39] R. Alterovitz, T. Siméon, and K. Goldberg, "The stochastic motion roadmap: A sampling framework for planning with markov motion uncertainty," in *Robotics: Science and Systems III*, 2008, pp. 233–241.

[40] K. Y. Goldberg, "Orienting polygonal parts without sensors," *Algorithmica*, vol. 10, no. 2, pp. 201–225, 1993. [Online]. Available: http://dx.doi.org/10.1007/BF01891840

[41] K. M. Lynch, M. Northrop, and P. Pan, "Stable limit sets in a dynamic parts feeder," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 608–615, Aug. 2002.

[42] T. D. Murphey, J. Bernheisel, D. Choi, and K. M. Lynch, "An example of parts handling and self-assembly using stable limit sets," in *Int. Conf. Int. Rob. Sys.*, Aug. 2005, pp. 1624–1629.

[43] O. C. Goemans, K. Goldberg, and A. F. van der Stappen, "Blades: a new class of geometric primitives for feeding 3d parts on vibratory tracks," in *Int. Conf. Rob. Aut.*, May 2006, pp. 1730–1736.

[44] T. H. Vose, P. Umbanhowar, and K. M. Lynch, "Friction-induced velocity fields for point parts sliding on a rigid oscillated plate," in *Robotics: Science and Systems*, Zurich, Switzerland, June 2008.

[45] S. Akella, W. H. Huang, K. M. Lynch, and M. T. Mason, "Parts feeding on a conveyor with a one joint robot," *Algorithmica*, vol. 26, no. 3, pp. 313–344, 2000.

[46] A. van der Stappen, R.-P. Berretty, K. Goldberg, and M. Overmars, "Geometry and part feeding," *Sensor Based Intelligent Robots*, pp. 259–281, 2002.

[47] K. Goldberg, B. V. Mirtich, Y. Zhuang, J. Craig, B. R. Carlisle, and J. Canny, "Part pose statistics: estimators and experiments," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 849–857, Oct. 1999.

[48] M. Moll and M. Erdmann, "Manipulation of pose distributions," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 277–292, 2002.

[49] S. Akella and M. T. Mason, "Using partial sensor information to orient parts," *The International Journal of Robotics Research*, vol. 18, no. 10, pp. 963–997, 1999. [Online]. Available: http://ijr.sagepub.com/cgi/content/abstract/18/10/963

[50] T. D. Murphey and J. W. Burdick, "Feedback control methods for distributed manipulation systems that involve mechanical contacts," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 763–781, 2004.

[51] H. L. Royden, *Real Analysis*, 3rd ed. Prentice Hall, 1988.

[52] J.-P. Laumond, "Finding collision-free smooth trajectories for a non-holonomic mobile robot," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 1987, pp. 1120–1123.

[53] J.-P. Laumond, P. Jacobs, M. Taix, and R. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 577–593, 1994.