Motion Planning under Bounded Uncertainty Using Ensemble Control

Aaron Becker
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
abecker5@illinois.edu

Timothy Bretl
Department of Aerospace Engineering
University of Illinois at Urbana-Champaign
tbretl@illinois.edu

Abstract—This paper considers the problem of motion planning for a nonholonomic unicycle despite uncertainty 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 the position of this ensemble is controllable, and derive motion planning algorithms to steer this position between a given start and goal. We apply our work to a differential-drive robot with unknown but bounded wheel radius, and validate our approach with hardware experiments.

I. INTRODUCTION

In this paper we consider the problem of motion planning for a nonholonomic unicycle in the presence of uncertainty that scales both the forward speed and the turning rate by an unknown but bounded constant. We focus on the unicycle because its kinematic model can be used to represent many different car-like vehicles of practical interest. We focus on bounded uncertainty of this special form—scaling the control inputs by an unknown constant—both because it serves as a useful pedagogical example for the development of new planning algorithms and because it is encountered in a surprising variety of real robotic systems.

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 $u_1 \in \mathbb{R}$ and the turning rate $u_2 \in \mathbb{R}$. Corresponding to these inputs, we define vector fields $g_1, g_2 \colon \mathcal{Q} \to T_q \mathcal{Q}$ by

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

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 start and goal configurations q_{start} and q_{goal}, the motion planning problem for a single unicycle is to find inputs

$$u_1(t) \colon [0, T] \to \mathbb{R}$$
$$u_2(t) \colon [0, T] \to \mathbb{R},$$

that result in $q(0) = q_{\text{start}}$ and $q(T) = q_{\text{goal}}$ for free final time T. Depending on the nature of the unicycle (e.g., car-like, differential-drive, etc.) these inputs might be subject to constraints like $|u_2(t)| \leq |u_1(t)| \leq 1$ that enforce a maximum speed and a minimum turning radius. Similarly, we might require that $q(t) \in \mathcal{Q}_{\text{free}}$ to consider collision-avoidance.

We will solve this same motion planning problem, but under uncertainty that can be captured by scaling both the forward speed u_1 and the turning rate u_2 by some unknown, bounded constant. The resulting kinematic model has 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 $0 \leq \delta < 1$. However, rather than try to steer a single robot governed by the uncertain kinematic model (2), our approach is to steer an uncountably infinite collection of robots parameterized by ϵ, each one governed by the exact kinematic model

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

Following the terminology introduced by recent work in control theory [1]–[7], we call this fictitious collection of robots an *ensemble* and call the model (3) an *ensemble control system*. The idea is that if we can find a single set of inputs $u_1(t)$ and $u_2(t)$ that result in $q(0, \epsilon) = q_{\text{start}}$ and $q(T, \epsilon) = q_{\text{goal}}$ for all $\epsilon \in [1 - \delta, 1 + \delta]$, then we can certainly guarantee that the actual robot, which corresponds to one particular value of ϵ, will move from start to goal.

We will begin in Section II with a brief review of ensemble control theory and other related work. In Section III, we will proceed to show that although the system (3) is not fully controllable, it is nonetheless controllable with respect to position—in other words, we can achieve any arbitrary position $x(T, \epsilon)$ and $y(T, \epsilon)$, even if we cannot achieve any arbitrary heading $\theta(T, \epsilon)$. Given that one can interpret what we are doing as steering an uncountably infinite number of unicycles under the constraint that every one of them receives the same control input, this controllability result may come as a bit of a surprise. Based on this result, we will derive a practical motion planning algorithm in Section IV that steers the system (2) from start to goal, regardless of ϵ. Finally, in Section V 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 motion planning with hardware experiments. We do not consider obstacle-avoidance within the scope of this paper—this and several other possible extensions will be discussed in our concluding remarks (Section VI).

II. RELATED WORK

A. Ensemble Control

Ensemble control, as presented in [1]–[7], extends the theory of nonlinear controllability from finite-dimensional systems, for example of the form (1) in Section I, to a particular class of infinite-dimensional systems characterized by a dispersion parameter, for example of the form (3) where this parameter is ϵ. Such an extension is necessary because infinite-dimensional systems typically evolve on configuration spaces that are not compact, a condition that is used in the proof of standard controllability theorems. Chow's theorem, for instance, implies that the drift-free system (1) is locally controllable because the vector fields 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 $q \in \mathcal{Q}$ (e.g., see [8]), but this theorem says nothing about the nearly identical system (3). To get around this problem, we 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. However, systems like ours are ignored by [7] after noting that the drift-free control system

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

is not controllable if $g_1, \ldots, g_m$ generate a nilpotent Lie algebra. We will show explicitly that (3) is not fully controllable in Section III, but we will then proceed to show that this system is nonetheless controllable with respect to position.

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 [9]. 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]–[7]. Robotics researchers are also beginning to adopt the term "ensemble," for example in the context of multi-robot formations [10], but the formal methodology of ensemble control has yet to be applied. We should also note that other approaches to dealing with infinite-dimensional systems (such as taking advantage of differential flatness) have been developed in parallel, as in [11].

B. 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 [12]–[14] and examples of which range from

early work on preimage backchaining [15] to very recent work on needle-steering using the stochastic motion roadmap [16]. As one example, we have drawn particular inspiration from work on sensorless manipulation [17]. 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 [18]. 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 [19], [20]. These two strategies have been applied to a much wider array of mechanisms such as vibratory bowls and tables [21], [22] or assembly lines [18], [23], [24], and have also been extended to situations with stochastic uncertainty [25], [26] and closed-loop feedback [27], [28]. 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 large numbers of objects at once.

III. ANALYSIS OF CONTROLLABILITY

A. Method of Approach

In this section we will establish controllability results for the systems (1) and (3), which we described in Section I but reprint here for convenience:

$$\dot{q}(t) = g_1(q(t))u_1(t) + g_2(q(t))u_2(t) \tag{1}$$
$$\dot{q}(t, \epsilon) = \epsilon\big(g_1\left(q(t, \epsilon)\right) u_1(t) + g_2\left(q(t, \epsilon)\right) u_2(t)\big). \tag{3}$$

In both of these expressions, the control vector fields g_1, g_2 are given by

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

In our presentation, we follow the general framework outlined by [8] (also see [14], [29], [30]). To begin, recall that any drift-free control system of the form

$$\dot{q}(t) = \sum_{i=1}^{m} g_i(q(t))u_i(t) \tag{4}$$

is *controllable* if for every $q_{\text{start}}, q_{\text{goal}} \in \mathcal{Q}$ there exist inputs $u_1(t), \ldots, u_m(t)$ such that if $q(0) = q_{\text{start}}$ then $q(T) = q_{\text{goal}}$ for some $T > 0$. We assume here that the configuration space $\mathcal{Q}$ is a smooth n-dimensional manifold and that $(u_1, \ldots, u_m) \in U$ for a constraint set $U \subset \mathbb{R}^m$ containing the origin in the interior of its convex hull.

In many drift-free control systems of interest, the number of inputs m is strictly less than the dimension n of the tangent space $T_q \mathcal{Q}$ at each configuration $q \in \mathcal{Q}$. For example, in our

system (1) the number of inputs is $m = 2$ and the number of dimensions is $n = 3$, reflecting the nonholonomic constraint

$$\dot{q}_1 \sin q_3 - \dot{q}_2 \cos q_3 = 0$$

that says a unicycle—rolling without slipping—cannot instantaneously move sideways. The situation is worse for the ensemble control system (3), where again $m = 2$ but $n = \infty$. The key question is if these differential constraints are integrable—in other words, if we are "stuck" on an m-dimensional manifold in $\mathcal{Q}$ or if, through a clever sequence of inputs, we can span the entire space.

For the finite-dimensional system (4), this question is answered by computing the Lie algebra generated by $g_1, \ldots, g_m$. Recall that the configuration reached from $q_0 \in \mathcal{Q}$ by applying the sequence of inputs $u_i = 1$, $u_j = 1$, $u_i = -1$, and $u_j = -1$, all for some small time t, is given approximately by the Taylor series

$$q(4t) = q_0 + t^2 \left(\frac{\partial g_j}{\partial q} g_i - \frac{\partial g_i}{\partial q} g_j \right) + O(t^3),$$

where we recognize the t^2 term as the Lie bracket

$$[g_i, g_j] = \frac{\partial g_j}{\partial q} g_i - \frac{\partial g_i}{\partial q} g_j.$$

This bracket is a new control vector field that can be followed approximately, although at a slower rate than either g_i or g_j. The Lie algebra is the linear span of all such vector fields, generated by repeated bracketing. If the Lie algebra has rank n, then a finite-dimensional drift-free control system of the form (4) is controllable [8]. As an example, the Lie algebra of (1) is given by the span of g_1, g_2, and

$$[g_1, g_2] = \begin{bmatrix} \sin q_3 \\ -\cos q_3 \\ 0 \end{bmatrix},$$

where we note that $[g_1, g_2]$ is the previously forbidden control vector field that is normal to the direction of rolling. This Lie algebra has rank 3 everywhere, so (1) is controllable.

For infinite-dimensional systems like (3), we cannot directly apply the Lie algebra rank condition because the configuration space is no longer compact. Instead, our approach will be to explicitly approximate any desired control vector field, in this case some smooth function $f(\epsilon) \in T_q\mathcal{Q}$ of $\epsilon \in [1-\delta, 1+\delta]$, by a polynomial approximation in powers of ϵ generated using a sequence of higher-order Lie bracket motions. This approach is inspired by the one taken in [1]–[7].

B. Finding a Controllable Subsystem

Unfortunately, the ensemble control system (3), as we have expressed it so far, is not controllable. This result was suggested by [3] and we will now prove it by construction. In particular, notice that for any u_1 and u_2 we have

$$\dot{q}_3(t, \epsilon) = \epsilon u_2(t).$$

As a consequence, if we define an auxiliary state $\gamma(t) \in S^1$ such that $\gamma(0) = 0$ and

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

then it is clear that

$$q_3(t, \epsilon) = q_3(0, \epsilon) + \epsilon \gamma(t)$$

for all $\epsilon \in [1-\delta, 1+\delta]$. In other words, the change in heading of each robot in the ensemble after the application of any input is a linear function of ϵ. This result implies that if

$$\frac{q_3(T, \epsilon) - q_3(0, \epsilon)}{\epsilon} \neq \text{constant},$$

as will certainly hold in general, then it is not possible to achieve any arbitrary $q_{\text{goal}} \in \mathcal{Q}$. Hence, the system (3) is not controllable.

However, this result also suggests the construction of a subsystem that, as we will show in the following section, is 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 have just introduced above. We will denote the configuration space by $\mathcal{P}$. We have just shown that the evolution of this subsystem is governed by the alternate kinematic model

$$\dot{p}(t, \epsilon) = \epsilon h_1 \left(p(t, \epsilon), \epsilon \right) u_1(t) + h_2 \left(p(t, \epsilon), \epsilon \right) u_2(t), \quad (5)$$

where

$$
h_1 \left(p(t, \epsilon), \epsilon \right) = \begin{bmatrix} \cos \left(q_3(0, \epsilon) + \epsilon p_3(t, \epsilon) \right) \\ \sin \left(q_3(0, \epsilon) + \epsilon p_3(t, \epsilon) \right) \\ 0 \end{bmatrix}
$$
$$
h_2 \left(p(t, \epsilon), \epsilon \right) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (6)
$$

and $q_3(0, \epsilon)$ is the initial heading given by q_{start}, as before. For convenience, we will abbreviate

$$c_3(t, \epsilon) = \cos \left(q_3(0, \epsilon) + \epsilon p_3(t, \epsilon) \right)$$
$$s_3(t, \epsilon) = \sin \left(q_3(0, \epsilon) + \epsilon p_3(t, \epsilon) \right)$$

so that

$$h_1 \left(p(t, \epsilon), \epsilon \right) = \begin{bmatrix} c_3(t, \epsilon) \\ s_3(t, \epsilon) \\ 0 \end{bmatrix}. \quad (7)$$

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 (5) is, in fact, controllable.

C. Controllability Via Polynomial Approximation

We will now show that the infinite-dimensional subsystem (5) with control vector fields defined by (6)-(7) is controllable.

Taking Lie brackets, we have

$$\begin{aligned}
[\epsilon h_1, h_2] &= \epsilon \left(\frac{\partial h_2}{\partial p} h_1 - \frac{\partial h_1}{\partial p} h_2 \right) \\
&= 0 - \epsilon \begin{bmatrix} 0 & 0 & -\epsilon s_3 \\ 0 & 0 & \epsilon c_3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \epsilon^2 \begin{bmatrix} s_3 \\ -c_3 \\ 0 \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
[[\epsilon h_1, h_2], h_2] &= 0 - \epsilon^2 \begin{bmatrix} 0 & 0 & \epsilon c_3 \\ 0 & 0 & \epsilon s_3 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
&= -\epsilon^3 \begin{bmatrix} c_3 \\ s_3 \\ 0 \end{bmatrix} \\
&= -\epsilon^3 h_1.
\end{aligned}$$

Let us define

$$h_3 = \begin{bmatrix} -s_3 \\ c_3 \\ 0 \end{bmatrix},$$

so that $[\epsilon h_1, h_2] = -\epsilon^2 h_3$. All possible control vector fields that can be generated by repeating this process to an arbitrary level of bracketing k can be written in the form

$$f(\epsilon) = c h_2 + \sum_{i=0}^{k} \left(a_i \epsilon^{2i+1} h_1 + b_i \epsilon^{2i+2} h_3 \right) \qquad (8)$$

for some freely chosen set of coefficients a_i, b_i, and c. Just as for finite-dimensional systems, a Taylor series analysis tells us that each of these control vector fields can be followed approximately by some sequence of inputs u_1 and u_2.

For a given configuration $p(t, \epsilon)$ and a given value of ϵ, the vector fields h_1, h_2, and h_3 are orthonormal, so that

$$h_1^T h_1 = h_2^T h_2 = h_3^T h_3 = 1$$

and

$$h_1^T h_2 = h_2^T h_3 = h_3^T h_1 = 0.$$

As a consequence, we may write (8) as a set of three scalar equations

$$\begin{aligned}
h_1^T f(\epsilon) &= \sum_{i=0}^{k} a_i \epsilon^{2i+1} \\
h_2^T f(\epsilon) &= c \\
h_3^T f(\epsilon) &= \sum_{i=0}^{k} b_i \epsilon^{2i+2}.
\end{aligned}$$

We have already established that p_3 is constant in ϵ, so that by necessity $h_2^T f(\epsilon)$ is also constant in ϵ. Furthermore, the Stone-Weierstrass Theorem [31] 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 $\epsilon \in [1 - \delta, 1 + \delta]$. As a corollary, as long as $\delta < 1$, it is possible to choose a_i and b_i so that

$$\begin{aligned}
h_1^T f(\epsilon) &\approx \sum_{i=0}^{k} a_i \epsilon^{2i+1} \\
h_3^T f(\epsilon) &\approx \sum_{i=0}^{k} b_i \epsilon^{2i+2},
\end{aligned}$$

with error vanishing in k. This result implies that we can get arbitrarily close to any desired control vector field $f(\epsilon) \in T_p \mathcal{P}$. We immediately conclude that the subsystem (5) is controllable. In other words, it is always possible to find inputs $u_1(t)$ and $u_2(t)$ that steer an uncountably infinite number of unicycles to any arbitrary position $x(T, \epsilon)$ and $y(T, \epsilon)$, even if the heading $\theta(T, \epsilon)$ must remain a linear function of ϵ.

IV. MOTION PLANNING ALGORITHM

In the previous section, we showed that the subsystem (5) is controllable (i.e., that the ensemble control system (3) is controllable with respect to position). Based on this result, we will now derive a motion planning algorithm that steers the system (5) from start to goal. We will assume that "start" and "goal" are given as continuous functions $p_{\text{start}}(\epsilon)$ and $p_{\text{goal}}(\epsilon)$ of the parameter ϵ. For our application of interest—where (5) captures the range of possible outcomes for a single vehicle— it will always be the case that these functions are constant and have the form

$$\begin{aligned}
p_{\text{start}}(\epsilon) &= (x_{\text{start}}, y_{\text{start}}, 0) \\
p_{\text{goal}}(\epsilon) &= (x_{\text{goal}}, y_{\text{goal}}, 0).
\end{aligned}$$

However, our motion planning algorithm does not depend on this structure, so we ignore it for now. Our general strategy will be to use piecewise-constant inputs of a particular form, as suggested by [32] and reviewed in [8], [29], [33].

Before proceeding, notice that the vector field h_1 in (6) may be expressed

$$h_1\left(p(t, \epsilon), \epsilon\right) = R(\epsilon) \begin{bmatrix} \cos \epsilon p_3 \\ \sin \epsilon p_3 \\ 0 \end{bmatrix}$$

where

$$R(\epsilon) = \begin{bmatrix} \cos q_3(0, \epsilon) & -\sin q_3(0, \epsilon) & 0 \\ \sin q_3(0, \epsilon) & \cos q_3(0, \epsilon) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

so if we apply the transformation

$$\begin{aligned}
p'_{\text{start}}(\epsilon) &= 0 \\
p'_{\text{goal}}(\epsilon) &= R(\epsilon)^T \left(p_{\text{goal}}(\epsilon) - p_{\text{start}}(\epsilon) \right),
\end{aligned}$$

then without loss of generality it is always possible to assume $q_3(0, \epsilon) = 0$ for all ϵ. For convenience here, we will also assume that inputs are constrained by $|u_1| \leq 1$ and

$|u_2| \leq 1$ but that there is no minimum turning radius, corresponding to the differential-drive vehicle that we will consider in Section V. This assumption is easily relaxed.

A. Motion Primitives with Piecewise-Constant Inputs

Now, for a non-negative integer i, a positive angle ϕ, and freely chosen $a_i', b_i' \in \mathbb{R}$, consider a motion primitive of the following form:

$$(u_1, u_2) = \begin{cases} (0,1) & 0 \leq t < i\phi \\ (1,0) & i\phi \leq t < i\phi + a_i' \\ (0,-1) & i\phi + a_i' \leq t < 3i\phi + a_i' \\ (1,0) & 3i\phi + a_i' \leq t < 3i\phi + a_i' + b_i' \\ (0,1) & 3i\phi + a_i' + b_i' \leq t \leq 4i\phi + a_i' + b_i'. \end{cases}$$

This primitive specifies the following sequence of actions:

1) turn left in place an angle $i\phi$
2) drive straight a distance a_i'
3) turn right in place an angle $-2i\phi$
4) drive straight a distance b_i'
5) turn left in place an angle $i\phi$.

We can think of this sequence as a simple piecewise-constant approximation to a sinusoidal input of magnitude $i\phi$. It is easy to show by direct calculation that the result, if applied to the system (5) and assuming that $p_3(0, \epsilon) = 0$, is to achieve

$$\Delta p(\epsilon) = \begin{bmatrix} (a_i' + b_i')\epsilon \cos \epsilon i\phi \\ (a_i' - b_i')\epsilon \sin \epsilon i\phi \\ 0 \end{bmatrix}.$$

With the input transformation

$$a_i' = \frac{a_i + b_i}{2}$$
$$b_i' = \frac{a_i - b_i}{2}$$

for freely chosen $a_i, b_i \in \mathbb{R}$, we can write this expression in the decoupled form

$$\Delta p(\epsilon) = \begin{bmatrix} a_i \epsilon \cos \epsilon i\phi \\ b_i \epsilon \sin \epsilon i\phi \\ 0 \end{bmatrix}.$$

Because primitives of this form leave p_3 invariant, we are free to concatenate them. In particular, the result after applying $k+1$ primitives with $i = 0, \ldots, k$, followed by a final turn in place through an angle c, is to achieve

$$\Delta p(\epsilon) = \begin{bmatrix} \sum_{i=0}^{k} a_i \epsilon \cos \epsilon i\phi \\ \sum_{i=0}^{k} b_i \epsilon \sin \epsilon i\phi \\ c \end{bmatrix}.$$

Just as in Section III, our problem has been reduced to function approximation. Denote the first two components of p_{goal}' by $x(\epsilon)$ and $y(\epsilon)$, respectively. We have assumed that both $x(\epsilon)$ and $y(\epsilon)$ are continuous, so we can select coefficients a_i and

b_i making the following series converge as $k \to \infty$:

$$x(\epsilon) \approx \sum_{i=0}^{k} a_i \epsilon \cos \epsilon i\phi \tag{9}$$

$$y(\epsilon) \approx \sum_{i=0}^{k} b_i \epsilon \sin \epsilon i\phi. \tag{10}$$

B. Numerical Computation

To actually compute a_i and b_i in (9)-(10), we solve a linear programming problem. It is clear that each set of coefficients may be computed independently, and we apply the same procedure in both cases, so for ease of exposition we will focus on a_i. First, we sample N values $\epsilon_1, \ldots, \epsilon_N$ equally spaced in $[1 - \delta, 1 + \delta]$. Then, we construct the matrix

$$W = \begin{bmatrix} \epsilon_1 & \epsilon_1 \cos \epsilon_1 1\phi & \ldots & \epsilon_1 \cos \epsilon_1 k\phi \\ \vdots & \vdots & & \vdots \\ \epsilon_N & \epsilon_N \cos \epsilon_N 1\phi & \ldots & \epsilon_N \cos \epsilon_N k\phi \end{bmatrix}$$

and the vector

$$v = \begin{bmatrix} x(\epsilon_1) \\ \vdots \\ x(\epsilon_N) \end{bmatrix},$$

so that (9) may be written in discrete form as

$$Wa \approx v. \tag{11}$$

To minimize the resulting path length, we want to find the vector of coefficients a minimizing $\|a\|_\infty$ subject to the constraint that $\|v - Wa\|_\infty \leq e$ for a given error tolerance $e > 0$. Introducing an auxiliary decision variable s, we can write this linear program in the standard form

$$\begin{aligned} \text{minimize} \quad & s \\ \text{subject to} \quad & -\mathbf{1}e \leq v - Wa \leq \mathbf{1}e \\ & -\mathbf{1}s \leq a \leq \mathbf{1}s. \end{aligned}$$

As we discussed above, some k always exists making this linear program feasible for any $e > 0$. By increasing the number of sampled points N, we can also guarantee by continuity of $x(\epsilon)$ that (11) is a good approximation of (9).

C. Results in Simulation

Figure 1 shows the results of applying our motion planning algorithm to an ensemble control system of the form (5) for which $\delta = 0.2$ and $p_{\text{goal}}(\epsilon) = (1, 0, 0)$. In this example, we specified the error tolerance $e = 10^{-4}$ and used $k = 6$ motion primitives. To generate a clean plot, we chose $N = 100$, but the results are nearly identical even with coarse sampling of ϵ, for example when $N = 10$. We claimed in the presentation of our planning algorithm that the minimum achievable error decreases with the number of motion primitives k. Figure 2 provides empirical evidence supporting this claim for the same example as in Fig. 1.
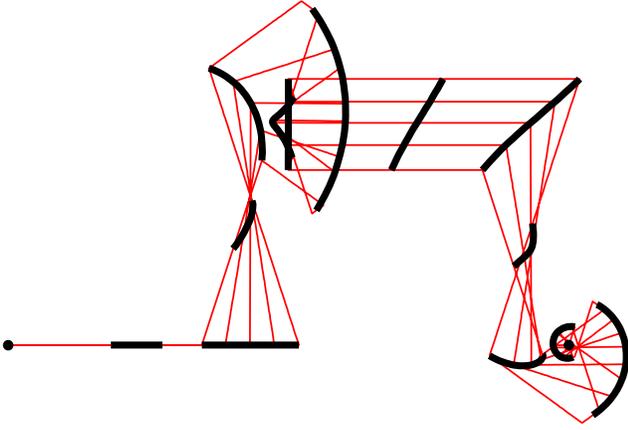
Fig. 1. An ensemble with $\epsilon \in [0.8, 1.2]$ moving a unit distance in the x direction, achieving an error tolerance of $e = 10^{-4}$. Thin lines show the path followed by robots with particular values of ϵ. Thick lines show the entire ensemble at instants of time.
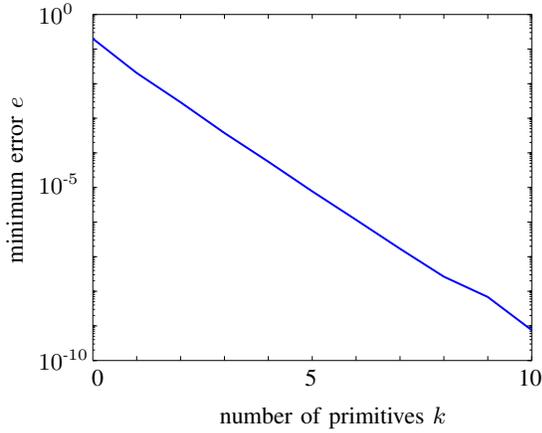
Fig. 2. For the example in Fig. 1, the minimum error e that can be achieved decreases with the number of motion primitives k.

V. HARDWARE EXPERIMENTS

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

A. Experimental Setup

Figure 3 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 without obstacles 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 $1°$.

Fig. 3. The differential-drive robot used for experimental validation.

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 [34]. 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. 4. 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 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 b, the forward speed and turning rate of a differential-drive robot are given by

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

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 $r \in [r_{\min}, r_{\max}]$. If we define

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

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

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

Fig. 4. We used four wheel sizes for experimental validation. These wheels are 10.16, 10.48, 12.7 and 15.24 cm in diameter.
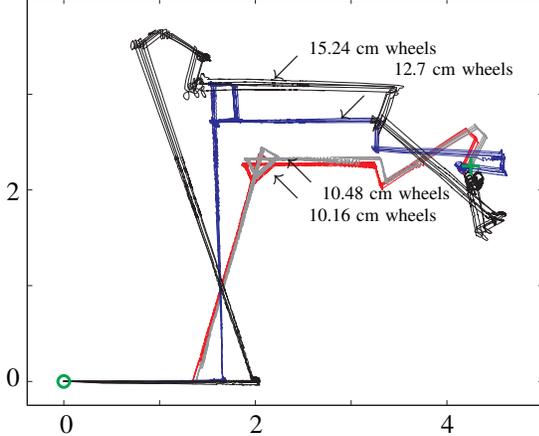
Fig. 5. Ground truth data gathered from camera system, starting at $(0, 0)$ with goal $(4.25, 2.25)$. Five runs for each wheel set are shown. Loops at the corners are artifacts from the camera system. Plots correspond to 10.16, 10.48, 12.7 and 15.24 cm wheels. Units are in meters.

Fig. 6. Ending position for each run. Large '+' for goal position, 'x' for expected ending position under zero odometry drift, 'o' for actual ending positions. The four symbol types correspond to 10.16, 10.48, 12.7 and 15.24 cm wheels. Units are in meters.

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

This expression simplifies if we select wheel angular velocities

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

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

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

and we have our result.

C. Experimental Results

Figure 5 shows the results of our experiments, which successfully validated our approach. We selected a goal location at $x = 4.25, y = 2.25$ m with the robot starting at the origin, and an error tolerance of 2cm. We applied the algorithm described in Section IV to plan a single sequence of inputs—i.e., single path—that will take the robot from start to goal regardless of its wheel size. Five runs were recorded for each wheel size. Fig. 5 shows the resulting trajectories, all of which reached a small neighborhood of the goal position.

The effects of dead-reckoning drift are obvious in Fig. 6. This drift is due to wheel slip, gear backlash, surface irregularities, wheel flex and other unmodeled disturbances. The vision system adds additional error. Ground truth position information was calculated from markings on the top of the robot. These markings were level and centered over the wheelbase for the 10.48 cm wheels, but tilted by $10°$ for the

largest wheels. These effects are most noticeable when the robot turns in place.

Table I shows the in-group mean and variance for the different wheel sets. The smallest variance is seen for the 10.48 wheels because the system was calibrated for these wheels. The 10.48 cm wheels are aluminum with rubber o-rings stretched over the rim, while the other wheels are 0.64 cm thick ABS plastic with a molded rubber traction ring covering the rim. The edge of the plastic wheels has a rectangular cross section, making the effective wheel base slightly larger than for the aluminum wheels. This causes larger errors on the ABS wheels. The means and variances tend to increase with the wheel diameter because the wheel diameter linearly scales all disturbance perturbations. The group mean error for distance is 0.098 m, five times larger than the specified terminal region, and the group variance is 0.004 m^2. The group mean error for heading is -0.007 rad and the group variance is 0.0001 rad^2.

VI. CONCLUDING REMARKS

In this paper we applied the framework of ensemble control theory to derive a motion planning algorithm that steers a nonholonomic unicycle from start to goal despite uncertainty that scales both the forward speed and the turning rate by an unknown but bounded constant. We validated our approach using a differential-drive robot with unknown but bounded wheel

radius and showed the results with hardware experiments.

Ensemble control theory is a recent development, and we feel that its application to robotics has raised a number of interesting questions for future work. For example, our algorithm constructs feasible inputs, but what are the optimal ones? We have shown controllability, but what about small-time local controllability that would allow us to deal in a systematic way with obstacles? (The fact that Fig. 1 simply scales with distance traveled indicates that small-time local controllability likely holds, but we require a proof.)

We would also like to address what may seem like an obvious flaw in our approach, particularly in the context of our chosen hardware demonstration. Within the scope of this paper, the plans we construct are paths. By ignoring the error that would result if a robot actually followed these paths, we are implicitly assuming the existence of a feedback control policy. Such a policy would require sensors, and these sensors might also be used to simply identify the unknown parameter ϵ, thus obviating the problem we consider. This objection is certainly reasonable, but there are two points to keep in mind. First, sensor data may be sufficiently noisy or come at a sufficiently low rate to make online system identification impractical while still allowing periodic course corrections (perhaps more intriguing, the feedback control policy might be sufficient to drive system uncertainty back to a 1-D manifold, but not to resolve position on this manifold). Second, although we focus here on planning for one robot with unknown ϵ, some applications—in particular, those that involve micro/nano-scale robotic systems—require planning for many robots, each with slightly different ϵ. In this latter case, online system identification is essential but does not remove the problem we consider. Regardless, this topic is an opportunity for future inquiry.

VII. Acknowledgements

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

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

[10] 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.

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

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

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

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

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

[16] 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.

[17] 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.

[18] 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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

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

[27] 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

[28] 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.

[29] J.-P. Laumond, Ed., *Robot Motion Planning and Control*, ser. Lecture Notes in Control and Information Sciences. Springer Berlin / Heidelberg, 1998, vol. 229/1998.

[30] H. Choset, K. Lynch, S. Hutchinson, G. Kanto, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

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

[32] 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.

[33] S. Sastry, *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.

[34] 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.