Feedback Control of Multiple Balls using One Plate-Ball System

Yu Huang and Aaron T. Becker

***Abstract*— In previous work we examined the classical plate-ball system w**

In this paper we revisit the classical plate-ball system and prove this system remains controllable under model perturbation that scales the ball radius by an unknown but bounded constant. We present an algorithm for approximate steering and validate the algorithm with hardware experiments. To perform these experiments, we introduce a new version of the plate-ball system based on magnetic actuation. This system is easy to implement and, with our steering algorithm, enables simultaneous manipulation of multiple balls with different radii.

I. Introduction

In previous work

[[insert references]]

we proved that a plate-ball system can be used in open-loop (no feedback) to steer a ball of unknown radius from an initial orientation to a desired orientation. The work relied on a proof of controllability inspired by work in ensemble control [[insert references]].

In this work, we show how this same proof of controllability can be used to derive feedback controllers that can simultaneously roll multiple spheres from arbitrary initial orientations to goal locations using a global input provided by a single moving plate system.

II. Prior work

Revisit the proof of controllability

insert proof

Reference other peoples work on ensemble control

III. Theory

This section describes our plate-ball system, the equations of motion, the error metrics, techniques for approximating rotation about the worlds z-axis, and the effects of rolling a sphere.

A. Review on Quaternions

A unit quaternion can represent either orientation or a rotation. The general format of a quaternion is

$$\mathbf{q} = a + u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k} \tag{1}$$

where those numbers a, u_x, u_y, u_z are all real numbers, and the part a is defined as the scalar part of the quaternion and $u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k}$ is defined as the vector part of the quaternion, which can denote an arbitrary vector in the R^3 space. For the purpose to using quaternions to express the rotation in R^3 space, all quaternions we use must be unit quaternions. Therefore, (1) can represent a rotation only if $\sqrt{a^2 + u_x^2 + u_y^2 + u_z^2} = 1$. similar to Euler Formula of complex numbers, the Euler Formula also applies to quaternions:

$$e^{\frac{\theta}{2}(u_x\mathbf{i}+u_y\mathbf{j}+u_z\mathbf{k})} = \cos(\frac{\theta}{2}) + (u_x\mathbf{i} + u_y\mathbf{j} + u_z\mathbf{k})\sin(\frac{\theta}{2}) \tag{2}$$

The vector part of a unit quaternion represents the rotation axis and θ is the rotation angle. A rotation about the world x-axis by angle θ is

$$\mathbf{q}_x(\theta) = \cos(\frac{\theta}{2}) + \mathbf{i}\sin(\frac{\theta}{2}) \tag{3}$$

Similarly, a rotation about the world y-axis by angle θ is

$$\mathbf{q}_y(\theta) = \cos(\frac{\theta}{2}) + \mathbf{j}\sin(\frac{\theta}{2}) \tag{4}$$

A rotation about the world z-axis is

$$\mathbf{q}_z(\theta) = \cos(\frac{\theta}{2}) + \mathbf{k}\sin(\frac{\theta}{2}) \tag{5}$$

A rotation about the unit axis $[v_x, v_y, v_z]^t$ is

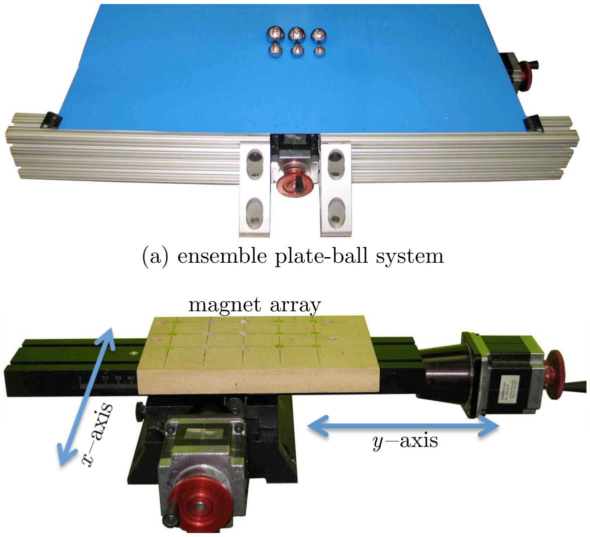
$$\mathbf{q}_{Arb}(\theta) = \cos(\frac{\theta}{2}) + (v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k})\sin(\frac{\theta}{2}) \tag{6}$$

Composing rotations with quaternions is represented by multiplication. A composite rotation formed by rotating $\mathbf{q}_1$ followed by $\mathbf{q}_2$ is

$$\mathbf{P}_1 = \mathbf{q}_2\mathbf{q}_1\mathbf{P}_0\mathbf{q}_1^*\mathbf{q}_2^* \tag{7}$$

Where $\mathbf{P}_1$ is the rotated vector or matrix and the $\mathbf{P}_0$ is the vector or matrix before rotation; $\mathbf{q}_1^*$ and $\mathbf{q}_2^*$ represent conjugate quaternions and the product happen between quaternions are all Hamilton Product. Note that traditional rotation matrices describe rotation using 9 numbers subscribed to 6 constraints whereas the quaternions describe it using 4 numbers subscribed to 1 constraints and can circumvent gimbal-lock and numerical instability. Therefore, in our algorithm simulation, we adopt quaternion rotation.

Yu Huang and A. Becker are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204-4005 USA {sshahrokhi2, atbecker}@uh.edu



(a) ensemble plate-ball system

(b) underlying mechanism

Fig. 1. The ensemble plate-ball system consists of an array of n metal spheres separated from an array of n magnets by a stationary sheet of cardboard. The magnet array is attached to an xy CNC table, and the spheres roll without slipping above their respective magnets. In this paper we prove this system, which applies the same control to each sphere, is approximately controllable. We then describe and implement controllers that approximately steer spheres of unknown diameters from a starting orientation to a goal orientation.

B. The plate-ball system

A plate ball system is composed of an array of n metal spheres separated from an array of n magnets by a stationary sheet of cardboard. The magnet array is attached to an xy CNC table, and the spheres roll without slipping above their respective magnets.

Equations for one sphere on a plate-ball system. Consider one ball with radius r_i whose Z-axis orientation after k steps‘s rotation is $\mathbf{Z}_{i,k}$. The plate-ball system can apply two inputs, a rolling movement in the x direction at the length of $l_{x,k+1}$:

$$\mathbf{Z}_{i,k+1} = \mathbf{q}_y(\frac{l_{x,k+1}}{r_i})\mathbf{Z}_{i,k}\mathbf{q}_y(\frac{l_{x,k+1}}{r_i})^{-1} \tag{8}$$

and a rolling movement in the y direction at the length of $l_{y,k+1}$:

$$\mathbf{Z}_{i,k+1} = \mathbf{q}_x(\frac{l_{y,k+1}}{r_i})\mathbf{Z}_{i,k}\mathbf{q}_x(\frac{l_{y,k+1}}{r_i})^{-1} \tag{9}$$

Obviously, this sphere cannot be rotated directly around the world z-axis. However, we can approximate z-rotation by Lie bracket motion.

We will proceed by showing how to construct $R_z\big(\hat{\theta}(\epsilon)\big)$. For small time dt, by rolling clockwise in a small square pattern on the plane, we can generate movement about the z-axis.

$$\begin{aligned}\exp\{\frac{-1}{\epsilon}\sqrt{dt}\Omega_y\}\exp\{\frac{-1}{\epsilon}\sqrt{dt}\Omega_x\}&\\ \cdot\exp\{\frac{1}{\epsilon}\sqrt{dt}\Omega_y\}\exp\{\frac{1}{\epsilon}\sqrt{dt}\Omega_x\}&\approx I+\epsilon^{-2}dt\Omega_z\\ &=\exp\{\epsilon^{-2}dt\Omega_z\}\end{aligned}$$

This Lie bracket movement generates the previously restricted motion about the z-axis.

An alternate method to approximate motion around the z-axis is to roll a sphere along a circular path starting at the sphere contact point. Consider a sphere with radius r_i centered at $(0,0,0)$ that rolls around a circular path of radius r_c centered at the point $(r_c\cos(\alpha_c), r_c\sin(\alpha_c), 0)$. The resulting motion can be decomposed into a rotation about an axis from $[r_c\cos(\alpha_c), r_c\sin(\alpha_c), 0]^\top$ to $[0,0,r_i]^\top$, followed by a rotation about the world z-axis of...

$$\psi_i = \cos(\theta) \tag{10}$$

If the sphere only rolls along an arc of length ℓ_c, the resulting rotation

$$\mathbf{q}(r_c, \alpha_c, \ell_c, r_i) = \cos(\theta) \tag{11}$$

For our purposes in this paper we want to align the z-axis of the sphere with the world z axis. The deviation of one sphere form the world z axis can be described in terms of $theta_i$ and ψ_i

The error metric for one sphere is derived by

$$\psi_i = \cos(\theta) \tag{12}$$

the error metric for multiple spheres is given by sum of squared error

$$\psi_i = \cos(\theta) \tag{13}$$

IV. Closed Loop Control

We start with a convenient mathematical fiction, and assume an input exists that rotates all spheres about the world z axis at the constant rate of $1/r_i$. Show that under this input the error metric (13) does not change.

A. A convenient fictitious input

We define a Lyapunov function

$$\psi_i = \cos(\theta) \tag{14}$$

with derivative

$$\psi_i = \cos(\theta) \tag{15}$$

We then select a control function that drives the Lyapunov function to 0.

$$\psi_i = \cos(\theta) \tag{16}$$

B. Steering a real system

We know that if we could rotate about the z axis with rotation rates $1/r_i.$, it doesn't change the error of the system We can then move after each roation to descrese our error We can continue until the system is at our goal state.

however, we cannot rotate about the z-axis.

We can roll the system along circular arcs, as shown by Eq. (11)

Our problem is then to find $(r_c, \alpha_c, \ell_c, r_i)$ that reduces (13). We then repeat this process until the error reduces to an acceptable level, or time runs out

Some plots of this.

Long description of your optimization function.

The plate-ball system is a canonical example of manipulation by rolling contacts [?]. In the classical version of this system, the ball is held between two parallel plates and manipulated by maneuvering the upper plate while holding the lower plate fixed. The ball can be brought to any position and orientation though translations of the upper plate.

We consider a variant of the plate-ball system in which the ball radius is an unknown but bounded constant. This variant has been considered previously by Oriolo et al. [?], who proposed a method of iterative feedback control. We are interested in the case where no sensor feedback is available.

Our main contribution is to prove this system is still controllable and present an algorithm for approximate (open-loop) steering. To do so, we apply the framework of ensemble control theory [?], [?], [?], [?] to derive an approximate steering algorithm. The basic idea, similar to early work on sensorless manipulation [?], is to maintain the set of all possible configurations of the sphere 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 [?] but involve approximations by repeated Lie bracketing that are reminiscent of seminal work on steering nonholonomic systems by Lafferriere and Sussman [?].

Our second contribution is a new experimental version of the plate-ball system based on magnetic actuation, shown in Fig. 1. This platform has several advantages over traditional plate-ball systems. The magnetic actuation makes the mechanism easy to implement, allows an unobstructed view of the ball on the plane, and enables simultaneous manipulation of multiple balls. If these balls have unique radii, by using our algorithm this system can steer each ball to arbitrary orientations.

The remainder of this paper proceeds as follows. We begin with a brief review of related work, focusing on sensorless part orientation and solutions to the plate-ball problem (Section V). We then present our problem variation (Section VI) and prove it is controllable (Section VII). We design an approximate steering algorithm (Section VIII), and validate the algorithm with a new hardware platform (Section IX). We discuss broader implications in our concluding remarks (Section X).

V. Related Work

A. Sensorless Part Orientation

We are motivated by progress in sensorless part manipulation, particularly the work of [?] and [?] showing that simple actuators are often sufficient to robustly orient a wide array of planar objects without using sensors. These works employed parallel-jaw grippers [?] and a tray that could be tilted in two axex [?]. These methods exploit differences in part geometry. Robustly orienting the rounded surface of a sphere offers special challenges due to its inherent symmetry.

B. Sphere Manipulation

Manipulation of spherical objects by rolling has been investigated in depth by members of the math, control, and robotic manipulation communities. This research can be traced to Brockett and Dai who analyzed an approximation of the problem and determined the optimal controller for this approximation [?]. Jurdjevic determined the optimal shortest length paths, showing that the optimal solution curve minimizes the integral of the geodesic curvature [?]. Li provided a symbolic algorithm for steering the system [?], while Marigo gave a numeric algorithm [?], and Oriolo and Vendittelli presented an iterative approach for stabilizing the plate-ball system [?].

This problem has produced several practical stabilizing controllers [?], [?]. Das and Mukherjee derive a motion planner for a plate-ball system with exponential stability [?].

Svinin and Hosoe extended the problem for plate-ball systems with limited contact area [?]. This enables manipulations of objects with spherical portions.

Several robotic plate-ball systems solutions have been implemented (see [?], [?]). Our approach using multiple balls is inherently underactuated, and in that respect is similar to Choudhury and Lynch's work that showed a single degree-of-freedom manipulator was sufficient for orienting a sphere. They designed a successful experiment consisting of an elliptical bowl mounted on top of a linear motor with the bowl's primary axis oriented 45 degrees from the linear motor orientation [?].

The control algorithms demonstrated in this work might find application at a much smaller scale using microspheres. Studies by Ding et al. on rolling friction of microspheres [?] demonstrate that even on the micro-scale spheres can roll with little slip. A study by Agayan et al. of the slipping friction of optically and magnetically manipulated microspheres on a glass-water interface demonstrate techniques for manipulation that may benefit from our methodology [?].

Lastly, in this paper we demonstrate that a continuum of differently-sized spheres is controllable. This means a finite number of spheres with heterogeneity in diameter are also controllable. Such a controller would be similar to the micro manipulation of Diller and Sitti et al. They controlled the

2D coordinates of multiple micro-scale permanent magnets by exploiting heterogeneity in the magnet's dimensions [**?**].

C. Ensemble Control

We are motivated by the work on *ensemble control* in [**?**], [**?**], [**?**], [**?**]. These works studied the controllability properties of the *Bloch equations*, a unit vector in $\mathbb{R}^3$. Much of this work can be applied to the sphere, which moves in $SO(3)$.

VI. Problem Statement

We are interested in approximate steering of the orientation of a sphere with unknown diameter by rolling on the plane.

A. One Sphere

Consider a sphere of radius r that rolls without slipping on a plane. Ignoring position, we describe its configuration by the rotation matrix R and its configuration space by $SO(3)$. The control inputs are the rolling speed along the x-axis u_1 and the rolling speed along the y-axis u_2. Corresponding to these inputs, we define a basis for the Lie algebra Ω_x, Ω_y where

$$\Omega_x = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}, \Omega_y = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \Omega_z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and write the kinematics of the sphere in the standard form

$$\dot{R}(t) = \frac{1}{r} R(t) \bigg(\Omega_x u_1(t) + \Omega_y u_2(t) \bigg). \tag{17}$$

Given $R_{\text{start}}, R_{\text{goal}} \in SO(3)$ and an error bound $\mu > 0$, the approximate steering problem is to find open-loop inputs

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

that result in $R(0) = R_{\text{start}}$ and $\operatorname{dist}(R(T), R_{\text{goal}}) \leq \mu$ for free final time T, where

$$\operatorname{dist}(R_a, R_b) = \arccos\left((\operatorname{trace}\left(R_a^{-1} R_b\right) - 1)/2 \right)$$

is the minimum angular distance between two rotation matrices. If such inputs always exist, then we say that (17) is *approximately controllable*—and indeed they do, since Ω_x, Ω_y, and the Lie bracket $[\Omega_x, \Omega_y]$ span the tangent space $T_R\left(SO(3)\right)$ everywhere.

B. Ensemble of Spheres

We will solve this same approximate steering problem, but under model perturbation of the sphere radius that scales the rolling speeds u_1 and u_2 by some unknown, bounded constant ϵ. The resulting kinematics have the form

$$\dot{R}(t) = \frac{\epsilon}{r} R(t) \bigg(\Omega_x u_1(t) + \Omega_y u_2(t) \bigg). \tag{18}$$

where $\epsilon \in [1-\delta, 1+\delta]$ for some $0 \leq \delta < 1$. Rather than try to steer one sphere governed by (18)—where ϵ is unknown—our approach is to steer an uncountably infinite collection of spheres parameterized by ϵ, each one governed by

$$\dot{R}(t, \epsilon) = \frac{\epsilon}{r} R(t, \epsilon) \bigg(\Omega_x u_1(t) + \Omega_y u_2(t) \bigg). \tag{19}$$

Following the terminology introduced by [**?**], [**?**], [**?**], we call this fictitious collection of spheres an *ensemble* and call (19) 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 $R(0, \epsilon) = R_{\text{start}}$ and $\operatorname{dist}(R(T, \epsilon), R_{\text{goal}}) \leq \mu$ for all $\epsilon \in [1-\delta, 1+\delta]$, then we can certainly guarantee that the actual sphere, which corresponds to one particular value ϵ^* of ϵ, will satisfy $\operatorname{dist}(R(T, \epsilon^*), R_{\text{goal}}) \leq \mu$. If such inputs always exist, then we say that (19) is *ensemble controllable*, interpreted as being approximately controllable on the function space $L_2\left([1-\delta, 1+\delta], SO(3)\right)$.

VII. Controllability

We will now prove that the system (19) 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 [**?**], and involves computations that are reminiscent of [**?**].

Theorem 1: The system (19) is ensemble controllable.

Proof: Any rotation A in $SO(3)$ can be parameterized by the rotations θ, ϕ, ψ about the world z-axis, x-axis and then z-axis.

$$A = R_z(\theta) R_x(\phi) R_z(\psi)$$

To approximate any rotation A with an ensemble, it is then sufficient to construct the rotations $R_z\big(\hat{\theta}(\epsilon)\big) \approx R_z(\theta)$, $R_x\big(\hat{\phi}(\epsilon)\big) \approx R_x(\phi)$ and $R_z\big(\hat{\psi}(\epsilon)\big) \approx R_z(\psi)$.

We will proceed by showing how to construct $R_z\big(\hat{\theta}(\epsilon)\big)$. For small time dt, by rolling clockwise in a small square pattern on the plane, we can generate movement about the z-axis.

$$\begin{aligned} &\exp\{\frac{-1}{\epsilon}\sqrt{dt}\Omega_y\}\exp\{\frac{-1}{\epsilon}\sqrt{dt}\Omega_x\} \\ &\quad \cdot \exp\{\frac{1}{\epsilon}\sqrt{dt}\Omega_y\}\exp\{\frac{1}{\epsilon}\sqrt{dt}\Omega_x\} \approx I + \epsilon^{-2} dt \Omega_z \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad = \exp\{\epsilon^{-2} dt \Omega_z\} \end{aligned}$$

This Lie bracket movement generates the previously restricted motion about the z-axis.

We take further Lie brackets to find new control vector fields:

$$\begin{aligned} \left[\frac{1}{\epsilon}\Omega_y, \frac{1}{\epsilon}\Omega_x\right] &= -\epsilon^{-2}\Omega_z \\ \left[\frac{1}{\epsilon}\Omega_y \left[\frac{1}{\epsilon}\Omega_y, \frac{1}{\epsilon}\Omega_x\right]\right] &= -\epsilon^{-3}\Omega_x \\ \left[\frac{1}{\epsilon}\Omega_y \left[\frac{1}{\epsilon}\Omega_y \left[\frac{1}{\epsilon}\Omega_y, \frac{1}{\epsilon}\Omega_x\right]\right]\right] &= \epsilon^{-4}\Omega_z \\ &\vdots \\ &= -1^k \epsilon^{-2k+1}\Omega_z \\ &= -1^k \epsilon^{-2k}\Omega_x. \end{aligned}$$

By successive Lie brackets, we can synthesize terms of the form $\{\epsilon^{-2k+1}\Omega_z\}$ and $\{\epsilon^{-2k}\Omega_x\}$. With these terms as generators, we can approximate the rotation

$$\exp\left\{\sum_{k=0}^{m} a_k \epsilon^{-2k+1}\Omega_z\right\},$$

The Stone-Weierstrass theorem [**?**] 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 $a \in \mathbb{R}^k$ so that

$$\theta(\epsilon) \approx \sum_{k=0}^{m} a_i \epsilon^{-2k+1}$$

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 $\epsilon^{-2k+1}\Omega_z$ and $\epsilon^{-2k}\Omega_x$, but remains finite for any given $\mu > 0$. Our result follows. ■

VIII. Controller Design

In this section we provide two methods for approximate open-loop steering of a rolling sphere, the first based on primitives that approximate an in-plane rotation of θ, and the second based on discrete optimization. The primitive-based method relies on concatenating primitives. Using fewer primitives reduces the cumulative error. It is well known that three orthogonal rotations span $SO(3)$. In the following section we show that two straight line rolls in the horizontal plane also span $SO(3)$. We then construct a motion primitive for spheres with radii $\in [1-\delta, 1+\delta]$ that approximates a rotation of θ about the world x-axis by a sequence of orthogonal rolls in the plane. This 2D path primitive can then be rotated an angle ψ about the z-axis by

$$\begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix}$$

to approximate a rotation about any axis parallel to the plane. Two of these primitives then are sufficient to approximately reorient all spheres with radii $\in [1-\delta, 1+\delta]$ by an arbitrary rotation in $SO(3)$.

A. Arbitrary Orientation in $SO(3)$ with Two Straight Rolls

Any rotation A can be described as a rotation about the orthogonal world reference frames $R_z(\theta)R_x(\phi)R_z(\psi)$. This corresponds to moving the sphere's north pole to latitude ϕ, longitude θ, then twisting about the original north pole ψ:

$$R_z(\theta)R_x(\phi)R_z(\psi) = \begin{bmatrix} c_\theta c_\psi - c_\phi s_\theta s_\psi & -c_\phi c_\psi s_\theta - c_\theta s_\psi & s_\theta s_\phi \\ c_\psi s_\theta + c_\theta c_\phi s_\psi & c_\theta c_\phi c_\psi - s_\theta s_\psi & -c_\theta s_\phi \\ s_\phi s_\psi & c_\psi s_\phi & c_\phi \end{bmatrix}$$

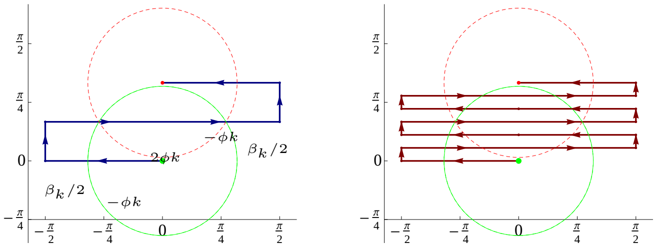


Fig. 2. A sample primitive of the form (20), with $\beta_k = \pi/3$ and $\phi k = \pi/2$. A sphere following this path generates to first order the rotation $\approx \beta_k \frac{1}{\epsilon}\cos(\phi k \frac{1}{\epsilon})$. By subdividing the rotation as illustrated on the right, we can improve arbitrarily this approximation, at the cost of a longer path.

Our system is constrained to roll along the plane. Any roll on the plane along a line of length ℓ making the angle α with the x-axis is given by

$$R_p(\ell, \alpha) = \begin{bmatrix} c_\alpha^2 + c_\ell s_\alpha^2 & (1-c_\ell)c_\alpha s_\alpha & s_\ell s_\alpha \\ (1-c_\ell)c_\alpha s_\alpha & c_\ell c_\alpha^2 + s_\alpha^2 & -c_\alpha s_\ell \\ -s_\ell s_\alpha & c_\alpha s_\ell & c_\ell \end{bmatrix}.$$

We can invert any rotation $A = R_z(\theta)R_x(\phi)R_z(\psi)$ with two straight line rolls in the plane of the form $A^{-1} = R_p(\pi, \frac{1}{2}(\theta - \psi)) \cdot R_p(\pi - \phi, \theta)$. The first movement rolls the sphere so that the original north pole is pointing down, and the second rotation returns the north pole to the original orientation. The first rotation is of length $\ell_1 = \pi - \phi$ with $|\phi| \leq \pi$ and the second rotation is of length $\ell_2 = \pi$, so the total path length is $\pi \leq \ell_1 + \ell_2 \leq 2\pi$.

B. Rotating Ensemble Approximately θ About the x-Axis

Consider the motion primitive given by Pryor in [**?**] of the following form for a non-negative integer k, and freely chosen $\phi, \beta_k \in \mathbb{R}$:

$$(u_1, u_2) = \begin{cases} (0,-1) & 0 \leq t < \phi k \\ (1,0) & \cdots \leq t < \cdots + \beta_k/2 \\ (0,1) & \cdots \leq t < \cdots + 2\phi k \\ (1,0) & \cdots \leq t < \cdots + \beta_k/2 \\ (0,-1) & \cdots \leq t < \cdots + \phi k \end{cases} \qquad (20)$$

Such a primitive is shown in Fig. 2. It is easy to show by direct calculation that the result is to achieve

$$\Delta x(\epsilon) = \beta_k,\ \Delta y(\epsilon) = 0,\ \Delta R(\epsilon) \approx R_y\left(\beta_k \frac{1}{\epsilon}\cos(\phi k \frac{1}{\epsilon})\right).$$

The approximation $R_y(\beta_k \frac{1}{\epsilon}\cos(\phi k \frac{1}{\epsilon}))$ is created by applying the motion primitive and taking the first-order Taylor series expansion at $\beta_k = 0$. The approximation degrades for large values of β_k, but by repeating the primitive (20) j times with parameter $\beta_k' = \beta_k/j$ the approximation improves. We can achieve arbitrary accuracy in the primitive by increasing j. Because primitives of this form all rotate in the same direction, we are free to concatenate them. The result after applying n primitives with $k = 1, \ldots, n$ is

$$\Delta x(\epsilon) = \sum_{k=1}^{n} \beta_k, \quad \Delta y(\epsilon) = 0, \quad \Delta R(\epsilon) \approx R_y\left(\Delta\theta(\epsilon)\right),$$

where we define $\Delta\theta(\epsilon) = \sum_{k=1}^{n} \beta_k \frac{1}{\epsilon}\cos(\phi k \frac{1}{\epsilon})$.

C. Achieving Error of a Particular Order

For convenience, we define $\epsilon^{-1} = 1+\delta$, and restrict $|\delta| < 1$. We may express the change in x and y after applying $n+1$ primitives, defined above, as a Taylor series in δ about $\delta = 0$:

$$\Delta\theta(\delta) = \Delta\theta(0) + \left.\frac{\partial \Delta\theta}{\partial \delta}\right|_{\delta=0} \delta + \frac{1}{2!} \left.\frac{\partial^2 \Delta\theta}{\partial \delta^2}\right|_{\delta=0} \delta^2 + \cdots .$$

This series has the form

$$\Delta\theta(\delta) = \sum_{j=0}^{n-1} s_j \delta^j + O\left(\delta^n\right).$$

Note that this expression is linear in s_j, and that each coefficient s_j is linear in $\beta_1, \dots, \beta_n$. Define

$$s = \begin{bmatrix} s_0 & \dots & s_{n-1} \end{bmatrix}^\top, \quad \beta = \begin{bmatrix} \beta_1 & \dots & \beta_n \end{bmatrix}^\top,$$

$$\text{so we can write } s = S\beta \tag{21}$$

for some matrix S that does not depend explicitly on δ. To achieve $\Delta\theta = 1$ with error that is of order n in δ, we require only that $s = [1, 0, \dots, 0]^\top$.

The achievable error decreases exponentially in the number $n+1$ of primitives used. We need to solve a system of n linear equations to achieve nth-order error. As a consequence, exactly n primitives are required to achieve nth-order error, for any n. There is an implicit assumption here that S is nonsingular (and, in practice, well conditioned), but this assumption will hold for "almost all" choices of ϕ.

By linearity, it is clear that the scaled parameters $\beta\Delta\theta$ will reach arbitrary $\Delta\theta$. We can use this method to generate paths that satisfy arbitrary accuracy bounds, but the required paths may be very long. Fig. **??** depicts paths generated to turn π about the x-axis with a threshold of $\pi/6$. In the following section we use optimal control to generate shorter paths.

D. Optimal Control with Piecewise-Constant Inputs

We want to steer an ensemble of spheres of radius $r\epsilon$ with primitives that consist of alternating rotations along the positive x- and y-axes to implement a desired rotation R_{goal}. The composite rotation $R(\Theta, \epsilon)$ is

$$R(\Theta, \epsilon) = \prod_{i=1}^{n} R_x\left(\frac{\theta_{2i-1}}{r\epsilon}\right) R_y\left(\frac{\theta_{2i}}{r\epsilon}\right) \tag{22}$$
$$\text{for } \Theta = [\theta_1, \dots, \theta_{2n}], \quad \theta_i \geq 0,$$

where $R_x(\alpha)$ is a rotation of α about the x-axis and $R_y(\beta)$ is a rotation of β about the y-axis. The problem is to find Θ such that the integral

$$\int_{1-\delta}^{1+\delta} \text{dist}\left(R\left(\Theta, \epsilon\right)\right), R_{goal}) \, d\epsilon \tag{23}$$

is minimized. In our discrete optimization routine, we replace the integral in (23) with a summation, and use gradient descent on Θ from multiple random seed values Θ_{init} to search for local minima in

$$\sum_{k=0}^{n-1} \text{dist}\left(R\left(\Theta, 1 - \delta + \frac{2\delta k}{n-1}\right), R_{goal}\right) \tag{24}$$

For the paths generated in Fig. **??**, we used 500 random seed values and $n = 15$. This optimization routine provided adequate results, but future work should take advantage of modern optimization methods.

IX. Hardware Implementation

A. System Design

The prototype spherical parts handler is shown in Fig. 1. A grid array of n magnets attached to a CNC xy-table can reorient n spheres by rolling in the plane when a nonmoving substrate is placed between the spheres and magnet array. A table-top CNC mill table is used as an xy planar manipulator to slide a tray in the plane. The tray is manufactured of MDF and is bolted to the mill. Fifteen holes are drilled into the surface. Each holds a 5×8 mm diameter cylindrical neodymium rare-earth magnet flush to the surface of the tray. The xy-table provides a 230×125 mm workspace. By securing a stationary, 0.15 mm thick cardboard sheet on top of the tray, we have a manipulator that can roll variable-sized ferro-magnetic spheres in the plane with minimal slip. Six spheres with diameters {16, 18, 19, 22, 24, 25.4} mm were used to test manipulation algorithms in this paper.

Our prototype was inspired by the kinetic sculptures of Bruce Shapiro and Jean-Pierre Hébert, which draw paths in sand using a rolling steel ball actuated by a hidden servo controlled magnet [**?**], [**?**].

B. Measuring System Position

Tracking the orientation of a sphere is a common problem, with approaches ranging from using colored circles painted at the vertices of Platonic solids [**?**], painting a fiducial pattern on a sphere and then comparing camera images to a library of generated rotated images [**?**], to tracking sphere orientation using dots precisely applied to the ball [**?**]. Each of these approaches requires accurate marker application, which increases in difficulty as the sphere shrinks in size. An alternate approach applies fiducial stickers to the sphere, learns their positions during a training phase, and then tracks these fiducials [**?**]. We adapted this method to use hand-drawn fiducials because stickers would affect the effective sphere diameter. We then estimated the sphere position by tracking five points in each image using Horn's method [**?**], producing at frame k the rotation matrix R_k. We used '⊤' shaped fidicials.

C. Results

1) Testing Movement Primitives: We compared motion primitives designed to rotate approximately π about the world y-axis. All tests were performed with six spheres ranging in size ±25% of nominal diameter. We compare primitives based on 1,2,3 and 4^{th}-order Taylor series with a primitive designed by discrete optimization over a 10 segment path. These movements are primitives because they can be concatenated to approximate any desired rotation. Note first that due to symmetry the rotation $R_y(\pi)$ is equivalent to rotating about any axis parallel to the plane π. Secondly, the Taylor series approximations can be linearly scaled in

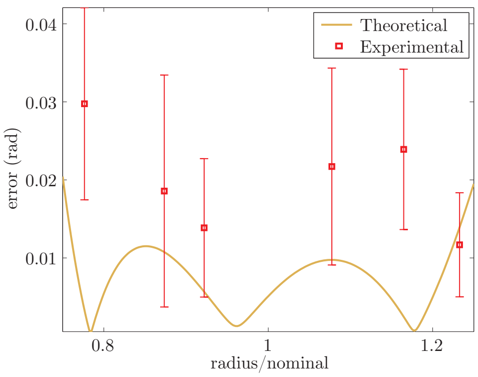


Fig. 3. Accuracy test for six spheres with diameters {16, 18, 19, 22, 24, 25.4} mm. The spheres were commanded to follow the 10-segment path in Fig. **??** designed by discrete optimization and the test was repeated 25 times. The yellow line is the theoretical error with no process noise. The red line shows the mean errors for each sphere tested ± one std.

the x-axis direction to approximate any rotation β about the world y-axis.

The paths for the Taylor series and the discrete optimization primitives are shown in Figs. **??** and **??**. These primitives were tested by measuring the starting and ending orientation for 10 iterations of the motion for the Taylor series primitives and 25 times for the discrete optimization path. The errors recorded for these tests are shown in Fig. **??** for all six sphere sizes.

For all spheres, the discrete optimization-based path performed best. The Taylor series-based paths increased in accuracy as the order increased from 1^{st} to 3^{rd} order, but the 4^{th} order path decreased in accuracy. The accuracy gained by the higher-order series approximation is cancelled by the process-induced drift shown in Fig. **??**.

X. Conclusion

We began with the problem of manipulating a plate-ball system despite model perturbation that scales the ball diameter by an unknown but bounded constant. We modeled the sphere as an ensemble control system, showed that this system is ensemble controllable, and derived an approximate steering algorithm to bring the sphere to within an arbitrarily small neighborhood of any given orientation in $SO(3)$. We applied our work to spheres with unknown but bounded diameters, and validated our approach with hardware experiments that simultaneously reoriented multiple spheres.

Our solutions consisted of open-loop paths that could be precomputed. We demonstrated that the hardware system has low noise that was ≈0.04% of commanded inputs. This noise introduces a drift term to the state evolution that cannot be countered by open-loop control. In many application environments, it is practical to add a camera system to sense the ball orientations. In this paper we showed that a continuum of different-sized spheres are approximately controllable by a shared input. Thus, a finite number of different-sized spheres is approximately controllable by a shared input. Using methods similar to [**?**] with the Bloch system, future work will apply feedback techniques to our plate-ball system.

Additionally, the classical plate-ball system is steerable in $\mathbb{R}^2 \times SO(3)$. Our controllability result of Section VII used bracket motion to reorient the spheres. These bracket motions are rotations in a square pattern on the plane, and leave the sphere x and y position unchanged. Therefore we can steer from a given $(x_{start}, y_{start}, R_{\text{start}})$ to a desired $(x_{goal}, y_{goal}, R_{\text{goal}}) \in [\mathbb{R}^2 \times SO(3)]$. We apply a straight line rotation to move the ensemble of spheres to x_{goal}, y_{goal}. This move incurs some rotation $\phi(\epsilon)$. We finish by performing a rotation in place that approximates $R_{\text{goal}}\phi(\epsilon)^\top R_{\text{start}}^\top$.

Finally, an important contribution of this paper was a new experimental platform, so we provided a characterization of the system reliability and described the system so it can be replicated. The magnetic actuation makes the mechanism easy to implement, allows an unobstructed view of the ball on the plane, and enables simultaneous manipulation of multiple balls with different radii.

XI. Acknowledgements

This work was supported by the National Science Foundation under CPS-0931871 and CMMI-0956362.