2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Motion Planning for Paramagnetic Microparticles Under Motion and Sensing Uncertainty

Wen Sun[1], Islam S. M. Khalil[2], Sarthak Misra[3], and Ron Alterovitz[1]

***Abstract*— Paramagnetic microparticles moving through fluids have the potential to be used in many applications, including microassembly, micromanipulation, and highly localized delivery of therapeutic agents inside the human body. Paramagnetic microparticles with diameters of approximately 100 μm can be wirelessly controlled by externally applying magnetic field gradients using electromagnets. In this paper, we introduce a motion planner to guide a spherical paramagnetic microparticle to a target while avoiding obstacles. The motion planner explicitly considers uncertainty in the microparticle's motion and maximizes the probability that the microparticle avoids obstacle collisions and reaches the target. To enable effective consideration of uncertainty, we use an Expectation Maximization (EM) algorithm to learn a stochastic model of the uncertainty in microparticle motion and state sensing from experiments conducted in a 3D 8-electromagnet microparticle testbed. We apply the motion planner in a simulated 3D environment with static obstacles and demonstrate that the computed plans are more likely to result in task success than plans based on traditional metrics such as shortest path or maximum clearance.**

I. Introduction

Paramagnetic microparticles are small particles (e.g., diameters around 100 μm) that can be controlled by externally applying a magnetic field gradient [1]–[3]. This type of micro-sized object is wirelessly actuated by electromagnets that are located around the workspace. By controlling the magnetic field gradient externally using electromagnets, the need to carry an on-board power source is eliminated, enabling the use of smaller particles like the one shown in Fig. 1. Due to their small size, paramagnetic microparticles have the potential to be used in a variety of applications, including microassembly, micromanipulation, and highly localized delivery of therapeutic agents inside the human body. [1], [3]–[5].

In this paper, we introduce a fast motion planner for guiding the motion of a spherical paramagnetic microparticle around obstacles to a target location in a static 3D fluid environment. Motion planning for paramagnetic microparticles is challenging for several reasons. First, motion plans must be feasible — the path should avoid obstacles and the required control inputs must satisfy constraints on maximum microparticle velocity and maximum currents sent to the electromagnets. Second, the motion plan should explicitly consider the substantial uncertainty in the motion of the microparticle and the sensing of system state. Sources of uncertainty that may cause the microparticle to deviate from its expected trajectory include noise in the electromagnet currents, noise in the electromagnetic field, variable flow rates, and noise in the image tracking.

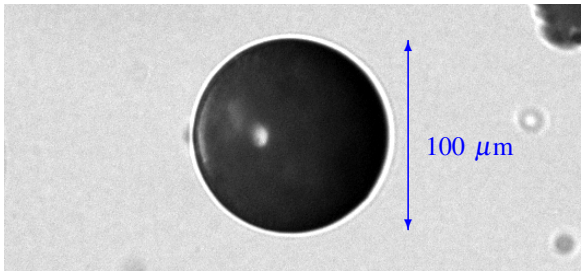


Fig. 1. A paramagnetic microparticle in a fluid can be wirelessly moved by controlling the surrounding magnetic field gradient using electromagnets. We introduce a motion planner that computes a path and a corresponding controller for the electromagnet currents to wirelessly guide the microparticle to a target location while avoiding obstacles.

Our motion planner computes a path and corresponding control policy (i.e., electromagnet currents) to guide the microparticle to a given target while avoiding collisions with specified obstacles. Our motion planner uses a sampling-based approach based on a Rapidly-Exploring Random Tree (RRT) [6], which guarantees that, if a solution exists, the solution will be found as computation time is allowed to increase. The motion planner is highly parallelizable and is sufficiently fast that we can generate hundreds of randomized feasible motion plans in a second and select the best plan based on a user-defined cost metric.

To address the challenge of uncertainty, we utilize cost metrics that are based on the probability of avoiding obstacles and reaching the target. In particular, we focus on two cost metrics: (1) maximizing the probability of success and (2) minimizing total path length subject to a chance constraint. These cost metrics require that we have a stochastic dynamics model of microparticle motion and sensing in order to *a priori* estimate the probability that the microparticle will avoid obstacles and reach the target when the plan is executed. We learn the parameters of the stochastic dynamics model by analyzing data from traces of experiments using an 8-electronmagnet 3D microparticle system [7]. Our learning method uses an Expectation-Maximization (EM) algorithm to learn the covariances of the position and velocity with respect to control inputs.

[1]Wen Sun and Ron Alterovitz are with the Department of Computer Science, The University of North Carolina at Chapel Hill, NC, USA. {wens, ron}@cs.unc.edu

[2]Islam S. M. Khalil is with German University in Cairo, New Cairo City, Egypt.

[3]Sarthak Misra is with MIRA-Institute for Biomedical Technology and Technical Medicine (Robotics & Mechatronics Group), University of Twente, The Netherlands. s.misra@utwente.nl

This research was supported in part by the National Science Foundation (NSF) through award IIS-1149965.

Our objective of explicitly considering uncertainty during motion planning has a substantial impact on our choice of the underlying motion planning algorithm. We build on the ideas of LQG-MP [8] and compute a Linear Quadratic Gaussian (LQG) controller for each RRT plan. Using the learned stochastic dynamics model, for each motion plan and corresponding LQG controller we use a fast, analytical method to quickly estimate the *a priori* probability of success of the microparticle motion plan. We can then select the best plan among the set of computed motion plans.

To the best of our knowledge, we present the first motion planner for microparticles that generates a collision-free path and considers the inherent uncertainty in microparticle motion and sensing. We apply the motion planner to a single microparticle operating in a simulated 3D environment that contains static obstacles. We evaluate the motion planner using a microparticle motion model learned from data from experiments using an 8-electromagnet microparticle system.

II. Related Work

Prior work on microparticles has made significant advances in controlling microparticle motion along paths [1]–[5]. For tasks in 2D settings, controllers have been introduced to perform contact and noncontact manipulation [9], to enable a single microparticle to follow given paths [3], and to control a cluster of paramagnetic microparticles for microassembly [10]. For tasks in 3D settings, controllers have been developed for demonstrating position and pose control of a mircorobot in an 8-electromagnet system [11], independent control of multiple microrobots in the same environment [12], and minimizing control effort for a single microparticle following a path [7]. Our focus is complementary to prior work on control; this paper focuses on computing plans that avoid obstacles.

Applying motion planning algorithms to microparticles to enable automatic obstacle avoidance is a new research challenge. Sampling-based motion planners such as RRT [6], RRT* [13], and probabilistic roadmaps (PRM) [6] have been successfully applied to different robots that have reliable (e.g., deterministic) dynamics. However, when planning for microparticles with uncertainty, optimal methods like RRT* and PRM cannot guarantee that solutions will approach optimality because the required optimal substructure property does not hold, i.e., the optimal path from a particular state is not independent of its prior history.

For motion planning under uncertainty, previous methods (e.g., [14]–[16]) can compute locally optimal plans along with associated control policies using frameworks based on partially observable Markov Decision Processes (POMDPs). Although these methods compute plans that avoid obstacles, they do not explicitly estimate the probability of success of a plan. Sampling-based methods (e.g., [17], [18]) have considered collision probabilities, but they also cannot accurately estimate the probability of success. In this paper, we utilize ideas from LQG-MP [8], which generates a plan and corresponding LQG controller for a robot and uses a metric correlated with collision probability. We also extend the method of [19] to use microparticle dynamics and quickly and explicitly estimate the probability of success, which includes both avoiding obstacles and reaching the goal.

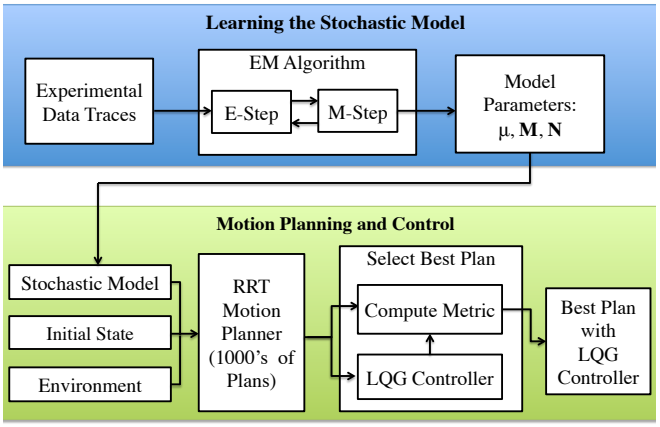


Fig. 2. Schematic overview of our method, including the learning of the stochastic model (Sec. IV) and the motion planning and control algorithms (Sec. V). The model learning only needs to be done once per microparticle and fluid, and then motion planning can be performed many times for different targets and obstacle locations.

III. Problem Definition

We consider motion planning for spherical paramagnetic microparticles with radius r_p on the order of 100 μm. The microparticles are composed of paramagnetic materials that have permanent magnetic moments, i.e, magnetic dipoles. Let $\mathbf{p} = [x, y, z]$ be the position of a microparticle in 3D space. We define the microparticle's state as $\mathbf{x} = [x, y, z, v^x, v^y, v^z]^T$, where v^x, v^y, and v^z specify the microparticle's velocity. We assume $\mathbf{p}$ can be sensed (subject to noise), e.g., by a vision system.

Microparticles move inside a fluid media under the influence of external magnetic field gradients. We assume n air-core electromagnets are located around the workspace. The magnetic force experienced by the microparticle is based on the superposition of each electromagnet's contribution. We control the gradient of the magnetic field by setting the current I_i applied to each electromagnet i, $i = 1, \ldots, n$. Hence, the control input is $\mathbf{u} = [I_1, \ldots, I_n] \in \mathcal{U}$ where $\mathcal{U} \subseteq \mathbb{R}^n$ is the set of feasible currents. We assume that continuous time τ is discretized into periods of equal duration Δ such that the t'th period begins at time $\tau = t\Delta$. We will select control inputs $\mathbf{u}_t$ to apply in each discrete period t.

The input to our motion planning approach is an estimate of the initial system state $\hat{\mathbf{x}}_0$ with corresponding variance Σ_0, the geometries of a set of obstacles $\mathcal{O}$, and a goal region $\mathcal{G}$ such that $\mathbf{x} \in \mathcal{G}$ signifies success. The objective of the motion planner is to compute a feasible path and corresponding linear feedback controller that guides the microparticle to the goal while avoiding obstacles. We focus on two uncertainty-based optimization objectives: (1) maximizing the probability of successfully reaching the goal without colliding with obstacles, and (2) minimizing path length subject to a chance constraint (e.g., a lower bound on the probability of success).

IV. Stochastic Model of Microparticle Motion

For efficient motion planning, we need a discrete-time stochastic model of microparticle motion and sensing. In this section, we formulate this model and describe how we learn the uncertainty parameters of the model from data traces from a 3D 8-electromagnet system as outlined in Fig. 2.

A. Modeling Microparticle Dynamics

Microparticles move under the influence of a magnetic force, a drag force, and a buoyancy force. We can control the magnetic force by applying currents to the magnets surrounding the microparticle's workspace. We define the magnetic force $\mathbf{f}(\mathbf{p}) = [F^x, F^y, F^z]^T \in \mathbb{R}^{3\times 1}$, where F^x, F^y, F^z are the components of $\mathbf{f}(\mathbf{p})$ in the x, y, and z directions, respectively. The magnetic force equals:

$$\mathbf{f}(\mathbf{p}) = \frac{4}{3\mu_0(1+\mathcal{X}_m)}\pi r_p^3 \mathcal{X}_m \mathbf{u}^T(\nabla(\tilde{\mathbf{B}}^T(\mathbf{p})\tilde{\mathbf{B}}(\mathbf{p})))\mathbf{u}, \quad (1)$$

where $\mathcal{X}_m$ is the magnetic susceptibility constant, μ_0 is the vacuum permeability, and $\tilde{\mathbf{B}} \in \mathbb{R}^{3\times n}$ is a matrix defining the magnetic field that depends on the position where the magnetic force is measured [11]. To compute currents for any desired magnetic force, we utilize a current-force map that allows us to invert equation 1 [7]. The gradient of the magnetic field is almost a constant in the workspace. Hence, we simply write the magnetic force as $\mathbf{f}$. We require $\mathbf{f} \in \mathcal{F}$, where $\mathcal{F}$ is the feasible set of control inputs. In our implementation, we require $|\mathbf{f}| \leq |\mathbf{f}|_{\max}$, where $|\mathbf{f}|_{\max}$ is the maximum norm of a magnetic force which can be generated by the system.

The drag force acting on the microparticle is $\mathbf{f}_d = [\kappa v^x, \kappa v^y, \kappa v^z]^T \in \mathbb{R}^{3\times 1}$, where $\kappa = -6\pi\eta r_p$ is a constant. We define the net buoyancy force acting on the microparticle as $F_b = V(\rho_b - \rho_f)g$, where V, ρ_b, and ρ_f are the volume and density of the microparticle, and the density of fluid, respectively. We also require $|\mathbf{v}| \leq |\mathbf{v}|_{\max}$ for some maximum feasible microparticle velocity $\mathbf{v}_{\max}$. Given $\mathbf{x}(\tau)$ and $\mathbf{f}(\tau)$ at time τ, we can define the continuous-time dynamics model of a microparticle as follows:

$$\dot{\mathbf{x}}(\tau) = \begin{bmatrix} v^x(\tau) \\ v^y(\tau) \\ v^z(\tau) \\ (F^x(\tau) + \kappa v^x(\tau))/m \\ (F^y(\tau) + \kappa v^y(\tau))/m \\ (F^z(\tau) - F_b + \kappa v^z(\tau))/m \end{bmatrix} \quad (2)$$

where m is the mass of the microparticle. We define matrix $\mathbf{A} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \frac{\kappa}{m}\mathbf{I}_{3\times 3} \end{bmatrix}$ and matrix $\mathbf{B} = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \frac{1}{m}\mathbf{I}_{3\times 3} \end{bmatrix}$. Hence, the above continuous-time dynamics can be written as:

$$\dot{\mathbf{x}}(\tau) = \mathbf{A}\mathbf{x}(\tau) + \mathbf{B}\mathbf{f}'(\tau) \quad (3)$$

where $\mathbf{f}'(\tau) = \mathbf{f}(\tau) + F_b[0,0,-1]^T$.

For purposes of motion planning and microparticle simulation, we need to derive the discrete-time dynamics based on the above continuous-time dynamics. We first note that $\mathbf{f}'(\tau)$ can serve as a high level control input for motion planning.

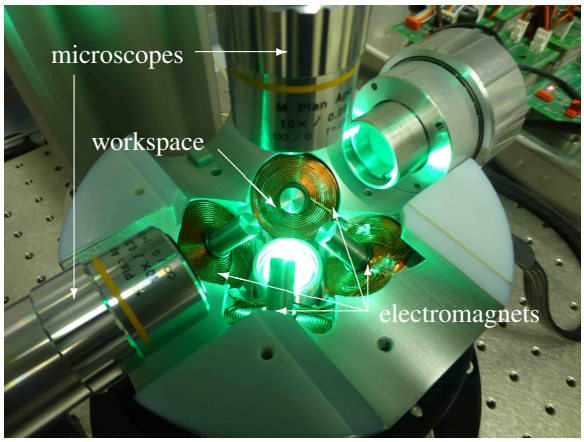


Fig. 3. Experimental testbed for learning the stochastic model. The system includes 8 iron-core electromagnets (only the bottom 4 are shown) that surround a $10 \times 10 \times 10$ mm^3 workspace. Inside the workspace we place a reservoir of fluid in which the microparticle can move. The system also includes 2 orthogonally placed microscopes for obtaining images of the workspace for the purpose of localizing the microparticle.

At any time step $t \in \mathbb{N}$ of duration Δ, the high level control input is defined as:

$$\mathbf{f}'_t = [F_t^x, F_t^y, F_t^z - F_b]^T, \ \mathbf{f}'_t \in \{\mathbf{f}' \mid |\mathbf{f}' - F_b\hat{\mathbf{n}}| \leq |\mathbf{f}|_{max}\}, \quad (4)$$

where $\hat{\mathbf{n}} = [0,0,-1]^T$. When the high level control input $\mathbf{f}'_t$ is constant between two successive time steps t and $t+1$, the state $\mathbf{x}_{t+1}$ can be computed given $\mathbf{x}_t$ by solving the differential equation 3:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{G}\mathbf{f}'_t, \quad (5)$$

with

$$\mathbf{F} = \exp(\mathbf{A}\Delta), \ \ \mathbf{G} = \int_0^\Delta \exp(\mathbf{A}(\Delta - \tau))\mathbf{B}\, d\tau, \quad (6)$$

where $\mathbf{F}$ and $\mathbf{G}$ can be evaluated analytically.

To model uncertainty, we assume that at any time step t, the dynamics is corrupted by additive noise $\mathbf{m}_t$ drawn from a Gaussian distribution with mean $\mu \in \mathbb{R}^{6\times 1}$ and variance $\mathbf{M} \in \mathbb{R}^{6\times 6}$:

$$\mathbf{x}_{t+1} = \mathbf{F}\mathbf{x}_t + \mathbf{G}\mathbf{f}'_t + \mathbf{m}_t, \ \ \mathbf{m}_t \sim \mathcal{N}(\mu, \mathbf{M}). \quad (7)$$

The tracking system is able to sense the microparticle's 3D position, and we assume the measurement is corrupted by noise $\mathbf{n} \in \mathbb{R}^3$ where $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{N})$ for some variance $\mathbf{N}$. This gives us the following sensing model:

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{n}) = \mathbf{C}\mathbf{x}_t + \mathbf{n}, \ \ \mathbf{C} = [\mathbf{I}_{3\times 3}, \mathbf{0}_{3\times 3}]. \quad (8)$$

The stochastic dynamics and the stochastic sensing model will be used to *a priori* estimate the probability of success of any feasible plan from the planner.

B. Learning the Uncertainty Parameters

In order to estimate the parameters μ, $\mathbf{M}$, and $\mathbf{N}$, we obtain traces of microparticle motion from a system [7] shown in Fig. 3. We recorded $K = 4$ independent trajectories. For each trajectory j, we derived and recorded a sequence of pairs $\{\mathbf{p}_t^j, \mathbf{u}_t^j\}$ where $1 \leq j \leq K, 0 \leq t \leq T_j$, $\mathbf{p}$ denotes the 3D position identified by the vision system, and T_j is the number

of time steps in trajectory j (which was several hundred for each trajectory). From $\mathbf{u}_t^j$ we can derive the $\mathbf{f}_t'^j$ using the modeling in Sec. IV-A.

The actual states $\mathbf{x}$ of the microparticle are latent. Similar to [20] except that we do not explicitly assume $\mu = \mathbf{0}$, we use the Expectation-Maximization (EM) algorithm to learn μ, $\mathbf{M}$, and $\mathbf{N}$ from the given data. We summarize the approach next.

a) E-Step: In the E-step, we assume μ, $\mathbf{N}$, and $\mathbf{M}$ are known. For each trajectory j, we use a Kalman smoother to compute the posterior distributions of all latent states $\mathbf{x}_t^j$ for $0 \leq t \leq T_j$. The Kalman smoother computes the Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_{t|t}^j, \Sigma_{t|t}^j)$, which is the distribution of $\mathbf{x}_t^j$ conditioned on all observations ($\mathbf{p}^j$) up to and including time step t. The Kalman smoother also computes a Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_{t+1|t}^j, \Sigma_{t+1|t}^j)$, which is the distribution of $\mathbf{x}_{t+1}^j$ conditioned on all observations up to and including time step t. Finally, the Kalman smoother returns the Gaussian distribution $\mathcal{N}(\hat{\mathbf{x}}_{t|T_j}^j, \Sigma_{t|T_j}^j)$, which is the posterior distribution of $\mathbf{x}_t^j$ conditioned on all observations in trajectory j.

b) M-Step: In the M-step, we assume the posterior distributions of all latent states $\mathbf{x}_t^j$ are available. We can model the expected likelihood of the recorded traces as:

$$\mathcal{Q}(\mu, \mathbf{M}, \mathbf{N}) = \mathbb{E}_{\mathbf{x}}(\log \prod_{j=1}^{K} \prod_{t=0}^{T_j} \mathbb{P}(\mathbf{p}_t^j | \mathbf{x}_t^j) \mathbb{P}(\mathbf{x}_{t+1}^j | \mathbf{f}_t'^j, \mathbf{x}_t^j)), \quad (9)$$

where

$$\mathbf{p}_t^j | \mathbf{x}_t^j \sim \mathcal{N}(\mathbf{C}\mathbf{x}_t^j, \mathbf{N}), \quad (10)$$

$$\mathbf{x}_{t+1}^j | \mathbf{f}_t'^j, \mathbf{x}_t^j \sim \mathcal{N}(\mathbf{F}\mathbf{x}_t^j + \mathbf{G}\mathbf{f}_t'^j + \mu, \mathbf{M}). \quad (11)$$

The expectation is taken with respect to the posterior distributions $\mathcal{N}(\hat{\mathbf{x}}_{t|T_j}^j, \Sigma_{t|T_j}^j)$ for $0 \leq t \leq T^j$, $1 \leq j \leq K$. We maximize the expected likelihood $\mathcal{Q}(\mu, \mathbf{M}, \mathbf{N})$ of the recorded traces with respect to μ, $\mathbf{M}$, and $\mathbf{N}$. We derived closed form update rules for updating μ, $\mathbf{M}$, and $\mathbf{N}$ by extending the approach of [20].

We start the E-step with an initial guess of μ, $\mathbf{N}$, and $\mathbf{M}$. We keep iterating the E-step and M-step until convergence. To properly model uncertainty of the microparticle system in our motion planner and simulation results in Sec. VI, we use the computed values for μ, $\mathbf{N}$, and $\mathbf{M}$ in conjunction with the discrete-time stochastic model in Sec. IV-A.

V. Motion Planning for Microparticles

In this section, we present a motion planner for microparticles that utilizes the stochastic motion and sensing model learned by the method in Sec. IV-B. As outlined in Fig. 2, we utilize RRT to generate a large set of feasible plans. For each feasible plan we compute a corresponding linear feedback controller (LQG) as in LQG-MP [8]. We then extend the method in [19] to estimate the probability of success of each plan for the microparticle. We select the best plan based on the chosen cost metric (e.g., maximizing probability of success or minimizing path length subject to a chance constraint). Once a plan is selected, we can execute the plan's corresponding linear feedback controller (LQG) to move the microparticle along the planned path.

A. Sampling-Based Motion Planning

For motion planning, we use a rapidly-exploring random tree (RRT), a well-established sampling-based motion planner that has been successfully used in a wide variety of applications [6]. The RRT is rooted at the microparticle's initial state $\hat{\mathbf{x}}_0$. At each iteration of the RRT algorithm, we sample a state $\mathbf{x}_{\text{sample}} \in \mathcal{Q}$, find its nearest neighbor $\mathbf{x}_{\text{near}}$ in the tree, and compute a feasible control $\mathbf{f}'$ that grows the tree from $\mathbf{x}_{\text{near}}$ toward $\mathbf{x}_{\text{sample}}$ using the RRT's "extend" function, i.e., $(\mathbf{x}_{\text{new}}, \mathbf{f}') = \text{extend}(\mathbf{x}_{\text{near}}, \mathbf{x}_{\text{sample}})$ where $\mathbf{x}_{\text{new}}$ requires no more than Δ time to reach from $\mathbf{x}_{\text{near}}$ [21]. The RRT's extend function uses the deterministic discrete-time dynamics model (Eq. 7 with $\mathbf{m}_t = \mu$). The output of the RRT is a nominal motion plan $\pi = [\mathbf{x}_0^\circ, \mathbf{f}_0'^\circ, \mathbf{x}_1^\circ, \mathbf{f}_1'^\circ, \ldots, \mathbf{x}_T^\circ, \mathbf{f}_T'^\circ]$, where T is the number of time steps and $\mathbf{x}_0^\circ = \hat{\mathbf{x}}_0$. We refer readers to [21] for details on RRT.

Instead of generating motion plans only from a single RRT structure (which has poor performance for optimization), our approach is to simultaneously execute a large number of independent RRTs to generate a set of feasible plans and then select the best plan based on the specified cost metric.

B. LQG Control for a Microparticle

The RRT planner provides a nominal plan π that satisfies $\mathbf{x}_{t+1}^\circ = \mathbf{F}\mathbf{x}_t^\circ + \mathbf{G}\mathbf{f}_t'^\circ + \mu$, for $0 \leq t \leq T-1$. To control the microparticle along the nominal plan π, we use the LQG controller since it provides optimal control for linear Gaussian motion and sensor models with a quadratic cost function penalizing deviation from the path. The LQG controller uses a Kalman filter for state estimation in conjunction with an LQR control policy [22].

Instead of directly controlling the state itself, we model deviation of the state with respect to the plan π. This is reasonable since our goal is to stay close to π. For $0 \leq t \leq T$, we define $\bar{\mathbf{x}}_t = \mathbf{x}_t - \mathbf{x}_t^\circ$, $\bar{\mathbf{f}}_t' = \mathbf{f}_t' - \mathbf{f}_t'^\circ$, and $\bar{\mathbf{z}}_t = \mathbf{z}_t - h(\mathbf{x}_t^\circ)$. Hence, the dynamics model and sensing model for deviations can be modeled as:

$$\bar{\mathbf{x}}_{t+1} = \mathbf{F}\bar{\mathbf{x}}_t + \mathbf{G}\bar{\mathbf{f}}_t' + \bar{\mathbf{m}}_t, \ \bar{\mathbf{m}}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{M}), \quad (12)$$

and

$$\bar{\mathbf{z}}_t = \mathbf{C}\bar{\mathbf{x}}_t + \mathbf{n}_t, \ \mathbf{n}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{N}). \quad (13)$$

In order to penalize deviations from π, we define the cost function as:

$$\mathbb{E}\left(\sum_{t=0}^{T} \bar{\mathbf{x}}_t^T \mathbf{C}\bar{\mathbf{x}}_t + \bar{\mathbf{f}}_t'^T \mathbf{D}\bar{\mathbf{f}}_t'\right). \quad (14)$$

These are the standard formulations of LQG control [22].

During execution, a Kalman filter is used to estimate the true deviation $\bar{\mathbf{x}}_t$ at each time step. We define the estimation of the state deviation at time step t as $\hat{\mathbf{x}}_t$, which can be obtained from the Kalman filter after sensing feedback is received. We then use an LQR controller for which we can compute the feedback matrix $\mathbf{L}_t$ for each time step. As the true state is unknown, we use the estimate $\hat{\mathbf{x}}_t$ from the Kalman filter to determine the next optimal control input. The control policy is then:

$$\mathbf{f}_t' = \mathbf{f}_t'^\circ + \mathbf{L}_t\hat{\mathbf{x}}_t. \quad (15)$$

The microparticle then will execute $\mathbf{f}_t'$ at time step t.

C. Optimization Objectives Based on Probability of Success

Both cost metrics (1) maximizing probability of success and (2) minimizing path length subject to a chance constraint require *a priori* estimation of the probability of success of a motion plan. With the stochastic dynamics and sensing of the microparticle, we extend the method in [19] to *a priori* estimate the probability of success of a feasible motion plan. This method *a priori* estimates the probability of collision of a motion plan assuming that a corresponding LQG controller is used during execution. Given a nominal plan π, $[\mathbf{x}_0, \mathbf{f}'_0, \ldots, \mathbf{x}_T, \mathbf{f}'_T]$, and an initial belief $\mathcal{N}(\hat{\mathbf{x}}_0, \hat{\Sigma}_0)$, we compute a sequence of Gaussian distributions $\{\mathcal{N}(\tilde{\mathbf{x}}_t, \tilde{\Sigma}_t)\}$, for $0 \leq t \leq T$. This sequence of Gaussian distributions captures the distributions of deviations from π during the execution of the plan with the LQG feedback controller.

To compute $\mathbb{P}_c$, the estimated probability of collision, we use the method in [19]. At time step t, let us assume the state of the microparticle is $\mathbf{x}_t \sim \mathcal{N}(\tilde{\mathbf{x}}_t, \tilde{\Sigma}_t)$. Before propagating to the time step $t+1$, we truncate this Gaussian distribution against obstacles to remove the parts of the distribution that collide with obstacles. The truncated distribution captures the possible states of the microparticle that are collision-free at time step t. By propagating the truncated distribution to time step $t+1$, we only propagate to the next time step states that are collision-free. Hence, we properly consider the dependency of uncertainty on previous time steps. Namely, the possible states of the microparticle at time step $t+1$ should be conditioned on the fact that the microparticle is collision-free at time step k, where $0 \leq k \leq t$.

We also need to estimate the probability that the microparticle reaches the goal region at the last time step. Given the belief of the microparticle at the last time step $\mathbf{x}_T \sim \mathcal{N}(\tilde{\mathbf{x}}_T, \tilde{\Sigma}_T)$, we can compute the probability of reaching the goal region as:

$$\mathbb{P}_g = \int_{\mathbf{p} \in \mathcal{G}} \frac{\exp(-\frac{1}{2}(\mathbf{p} - \Lambda\tilde{\mathbf{x}}_T)^T (\Lambda\tilde{\Sigma}_T\Lambda^T)^{-1}(\mathbf{p} - \Lambda\tilde{\mathbf{x}}_T))}{\sqrt{(2\pi)^3 |\Lambda\tilde{\Sigma}_T\Lambda^T|}} d\mathbf{p}, \tag{16}$$

where $\mathcal{G}$ is the goal region and $\Lambda = [\mathbf{I}_{3\times3}, \mathbf{0}_{3\times3}]$. In our implementation, we numerically estimate Eq. 16.

The probability of success can be computed as $\mathbb{P}_s = (1 - \mathbb{P}_c)\mathbb{P}_g$. Using the computations described above, we can evaluate the probability of success for each feasible plan and select the best plan based on the chosen cost metric.

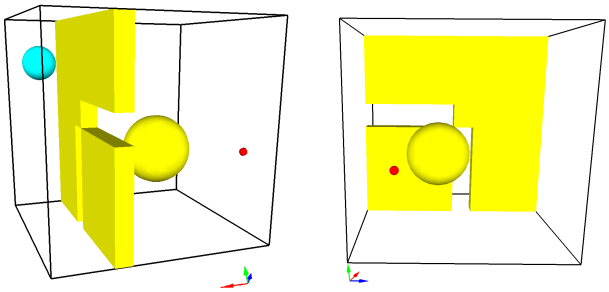
VI. Evaluation

We apply our motion planner in simulation to a microparticle moving in a 3D environment which contains multiple obstacles and a goal region. We evaluate our method for two cost metrics: (1) maximizing probability of success, and (2) minimizing path length subject to a chance constraint. We define a chance constraint as $\mathbb{P}_s \geq P_s$, where $0 \leq P_s \leq 1$ is specified by the user. The chance constraint imposes a lower bound on the probability of success (i.e., avoiding obstacles and reaching the goal region). We tested our C++ implementation on a PC with a 3.7 GHz Intel Core i7 processor.

Parameter	Value	Parameter	Value
r_p	50×10^{-6} m	μ_0	$4\pi \times 10^{-7}$ T.m /A
η	1 mPa.s	$\mathcal{X}_m$	0.17 ± 0.007
m	$7.33 \times 10^{-10} kg$	Δ	0.5 second
$\lvert\mathbf{v}\rvert_{\max}$	0.3 mm/s	$\lvert\mathbf{f}\rvert_{\max}$	3.0 nN

TABLE I

Experimental parameters [3], [23], where r_p and m are the radius and mass of the microparticle, respectively. We define $|\mathbf{v}|_{max}$ and $|\mathbf{f}|_{max}$ as the maximum velocity of the microparticle and the maximum magnitude of the magnetic force.



(a) Environment side view (b) Environment front view

Fig. 4. The environment workspace is 5cm × 5cm × 5cm. The objective is to move the microparticle from its initial position (red dot) to the goal region (cyan sphere) while avoiding obstacles (yellow). There are two narrow passages through the center obstacle; the passages are of equal length but the horizontal passage is 20% wider than the vertical passage. The width of the horizontal passage is 0.6 mm while the width of the vertical passage is 0.5 mm.

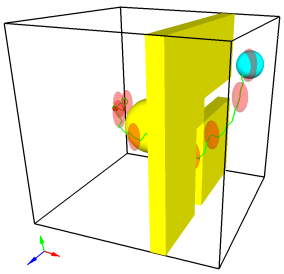
We use the method introduced in Section V-C to estimate the probability of success of a plan within our motion planner. To evaluate the true probability of success of a plan, we executed 1,000 Monte Carlo simulations in which we execute the plan's LQG controller in the environment with noise simulated using the learned noise models of Sec. IV-B. (We note the Monte Carlo simulations take substantially longer, and we do this to accurately assess our motion planner's performance.)

Although our experiments are conducted in simulation, we use the parameters from a physical magnetic prototype system [3], [23] and use the stochastic model learned in Sec. IV-B to simulate microparticle motion in the environment. Table I shows parameters for modeling the magnetic prototype system. Note that $|\mathbf{v}|_{\max}$ and $|\mathbf{f}|_{\max}$ are used as the constraints on the velocity and the control inputs in the RRT.

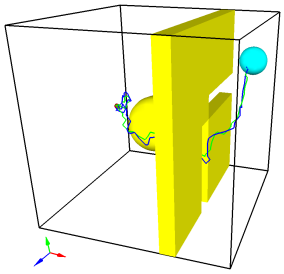
A. Maximizing Probability of Success

We first evaluate our motion planner with the cost metric of maximizing the probability of success for the environment in Fig. 4. Note that the horizontal narrow passage is 20% wider than the vertical narrow passage.

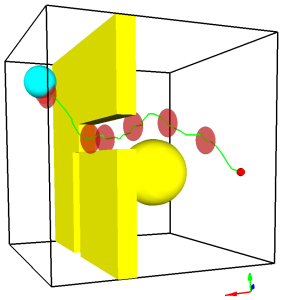
For evaluation, we executed the algorithm for 10,000 RRTs, which requires around 500 seconds for generating the RRTs and 50 total seconds to evaluate the probability of success metric for all plans.



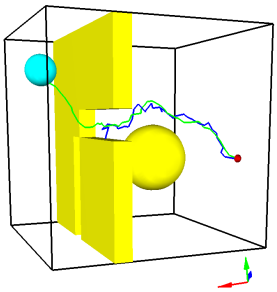
(a) Maximizing probability of success



(b) LQG control of the plan with highest probability of success



(c) Maximum Clearance



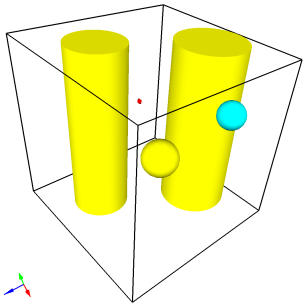
(d) LQR control of the plan with maximum clearance

Fig. 5. Our method computes a motion plan and corresponding controller with the objective of maximizing probability of success. For this environment, we show the computed nominal plan (green trajectory) (a). The red ellipses show the Gaussian distributions that capture the deviations of states along the nominal motion plan. We also show an example successful execution of LQG control (blue trajectory) along the nominal plan (b). The plan computed with the maximum clearance metric passes through the horizontal passage, which is wider than the vertical passage but not aligned with the microparticle's uncertainty (c). When executing a maximum clearance trajectory using the corresponding LQR controller, the microparticle is more likely to collide with an obstacle (blue example execution) (d) than when using the probability of success metric.

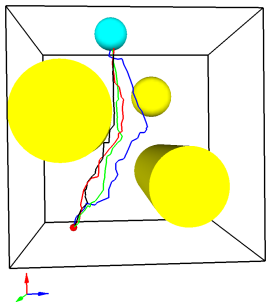
Fig. 5(a) shows the motion plan computed for maximizing the probability of success. The plan is shown in green, and the red ellipses show the estimated uncertainty distribution in position (3 standard deviations) at several time steps along the path. The estimated probability of success for the selected plan (using the fast analytical methods in Sec. V-C) is 96.1%. The true probability of success from Monte Carlo simulations is 99.6%. In Fig. 5(b) we illustrate an example execution of the LQG controller along the planned path while subject to noise from the model learned in Sec. IV-B.

To maximize the probability of success, the microparticle passes through the vertical passage. Although the horizontal passage is wider, the uncertainty model learned in Sec. IV-B has higher uncertainty in the Z direction, likely due to noise in the buoyancy force. The impact of the extra uncertainty can be seen in red ellipses in Fig. 5(a). Thus, it is safer to pass through the narrower vertical passage.

We compare our motion planning results using the probability of success metric to using a classical cost metric that is also related to safe motion planning: maximizing clearance in conjunction with an LQR controller, which does not require stochastic model learning (Sec. IV-B) or probability of success estimation (Sec. V-C). Fig. 5(c) shows the motion



(a) Initial environment



(b) Optimal plans for different cost metrics

Fig. 6. We evaluate the motion planner for cost metrics related to shortest paths in this environment (a) where the objective is to move the microparticle from its initial position (red dot) to the goal region (cyan sphere) while avoiding obstacles (yellow). We show (b) the computed plans for three cost metrics: minimizing path length (black), minimizing path length subject to C_1 (red), minimizing path length subject to C_2 (green) and maximizing clearance (blue).

plan computed for maximizing clearance from obstacles. To maximize clearance, the microparticle passes through the horizontal passage since it is wider. However, the larger uncertainties along the Z direction affect the probability of success. As shown in Fig. 5 (c), the uncertainty ellipses collide with the obstacles while passing through the horizontal passage. In addition, the cost metric of maximizing clearance aims for the goal region but does not explicitly consider the likelihood that the microparticle will end up inside the goal region due to uncertainty. The impact of these factors is that the true probability of success of the maximum clearance plan is 91%, substantially lower than the plan that explicitly maximizes probability of success.

B. Shortest Path Subject to a Chance Constraint

We also evaluated our method on another cost metric: minimizing path length subject to a chance constraint. We consider two chance constraints: C_1: $\mathbb{P}_s \geq 0.99$ and C_2: $\mathbb{P}_s \geq 0.90$. We compare to two classical cost metrics: minimizing path length and maximizing clearance, which are both used in conjunction with an LQR controller and do not require the stochastic model learning (Sec. IV-B) or probability of success estimation (Sec. V-C).

For evaluation, we executed the algorithm for 20,000 RRTs, which required around 200 seconds for generating the RRTs and 160 total seconds to evaluate the probability of success metric for all plans. As before, for shortest path and maximizing clearance, the true probability of success is evaluated using Monte Carlo simulations with LQR control.

Fig. 6 shows the results of using four different cost metrics. The black plan from minimizing path length also passes closest to an obstacle (the left cylinder in Fig. 6(b)). The red plan is the plan that moves furthest from the obstacle, such that it can satisfy C_1 : $\mathbb{P}_s \geq 0.99$. The blue plan for maximizing clearance passes through the middle of the narrow passage to achieve the largest clearance.

Table II shows the statistics for the four different cost metrics. As we can see, although the shortest path cost metric

Cost Metric	Length (mm)	Clearance (mm)	$\mathbb{P}_{est}$	$\mathbb{P}_{true}$
Shortest Path	5.66	0.02	/	37.9%
Max Clearance	6.70	0.53	/	86.9%
Shortest Path s.t C_2	5.79	0.12	94.0%	95.9%
Shortest Path s.t C_1	5.84	0.22	99.9%	99.8%

TABLE II

STATISTICS FOR FOUR COST METRICS. WE DENOTE $\mathbb{P}_{est}$ AS THE ESTIMATED PROBABILITY OF SUCCESS AND $\mathbb{P}_{true}$ AS THE TRUE PROBABILITY OF SUCCESS. LQR CONTROL IS APPLIED TO SHORTEST PATH AND MAX CLEARANCE METRICS. C_1 STANDS FOR $\mathbb{P}_s > 0.99$ AND C_2 STANDS FOR $\mathbb{P}_s > 0.9$.

finds the shortest path, it passes too close to the obstacles (clearance equals to 0.02 mm) to achieve a reasonable probability of success. On the other hand, our method using the shortest path subject to a chance constraint generates plans that enforce a minimum probability of success requirement. For a less restrictive chance constraint (C_2), we find a shorter path but with smaller probability of success. Compared to maximizing clearance, our method indeed not only finds a shorter path, but also achieves higher probability of success. This is because the cost metric of maximizing clearance does not explicitly consider the likelihood of reaching the goal. Our method can compute short paths while still maintaining high probability of success.

VII. CONCLUSION

In this paper, we introduced a motion planner to guide a spherical paramagnetic microparticle to a target while avoiding obstacles. The motion planner computes a path and a corresponding controller for the electromagnets' currents. Our motion planner explicitly considers uncertainty in the microparticle's motion; we formalized and utilized cost metrics that consider the probability that the microparticle avoids obstacle collisions and reaches the target. To enable effective consideration of uncertainty, our cost metrics relied on a stochastic model of the uncertainty in motion and sensing. The stochastic model was learned from experiments conducted in a 3D 8-electromagnet microparticle testbed.

We applied the motion planner in a simulated 3D environment with static obstacles and demonstrated that the computed plans are more likely to result in task success than plans based on traditional metrics such as shortest path or maximum clearance. In future work, we plan to integrate the motion planner with a 3D paramagnetic microparticle system to further evaluate algorithm performance. We also plan to investigate extending the motion planner to properly handle environments with moving obstacles.

VIII. ACKNOWLEDGMENT

The authors thank Bart Reefman for providing input regarding the experimental setup.

REFERENCES

[1] J. J. Abbott, Z. Nagy, F. Beyeler, and B. J. Nelson, "Robotics in the small, part I: microbotics," *IEEE Robotics and Automation Magazine*, vol. 14, no. 2, pp. 92—-103, Jun. 2007.

[2] M. Sitti, "Microscale and nanoscale robotics systems: characteristics, state of the art, and grand challenges," *IEEE Robotics and Automation Magazine*, pp. 53–60, Mar. 2007.

[3] I. S. M. Khalil, J. D. Keuning, L. Abelmann, and S. Misra, "Wireless magnetic-based control of paramagnetic microparticles," *Proc. IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)*, pp. 460–466, Jun. 2012.

[4] B. J. Nelson, I. K. Kaliakatsos, and J. J. Abbott, "Microrobots for minimally invasive medicine," *Annual Review of Biomedical Engineering*, vol. 12, pp. 55–85, Aug. 2010.

[5] M. Sitti, "Voyage of the microrobots," *Nature*, vol. 458, pp. 1121–1122, Apr. 2009.

[6] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[7] I. S. M. Khalil, R. M. P. Metz, B. A. Reefman, and S. Misra, "Magnetic-based minimum input motion control of paramagnetic microparticles in three-dimensional space," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 2053–2058.

[8] J. van den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *Int. J. Robotics Research*, vol. 30, no. 7, pp. 895–913, Jun. 2011.

[9] C. Pawashe, S. Floyd, E. Diller, and M. Sitti, "Two-dimensional autonomous microparticle manipulation strategies for magnetic microrobots in fluidic environments," *IEEE Trans. Robotics*, vol. 28, no. 2, pp. 467–477, Apr. 2012.

[10] I. Khalil, F. Brink, O. S. Sukas, and S. Misra, "Microassembly using a cluster of paramagnetic microparticles," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2013, pp. 5507–5512.

[11] M. P. Kummer, J. J. Abbott, B. E. Kratochvil, R. Borer, A. Sengul, and B. J. Nelson, "OctoMag: An electromagnetic system for 5-DOF wireless micromanipulation," *IEEE Trans. Robotics*, vol. 26, no. 6, pp. 1006–1017, Dec. 2010.

[12] E. Diller, J. Giltinan, and M. Sitti, "Independent control of multiple magnetic microrobots in three dimensions," *Int. J. Robotics Research*, vol. 32, no. 5, pp. 614–631, May 2013.

[13] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Research*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[14] R. Platt Jr., R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Robotics: Science and Systems*, 2010.

[15] M. P. Vitus and C. J. Tomlin, "Closed-loop belief space planning for linear, Gaussian systems," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 2152–2159.

[16] J. van den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *Int. J. Robotics Research*, vol. 31, no. 11, pp. 1263–1278, Sep. 2012.

[17] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 723–730.

[18] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robotics Research*, vol. 31, pp. 1263–1278, Sep. 2009.

[19] S. Patil, J. van den Berg, and R. Alterovitz, "Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2012, pp. 3238–3244.

[20] A. Coates, P. Abbeel, and A. Ng, "Learning for control from multiple demonstrations," in *Proc. Int. Conf. Machine Learning (ICML)*, 2008, pp. 144–151.

[21] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, B. R. Donald and Others, Eds. Natick, MA: AK Peters, 2001, pp. 293–308.

[22] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 2nd ed. Belmont, MA: Athena Scientific, 2000.

[23] I. S. M. Khalil, R. M. P. Metz, L. Abelmann, and S. Misra, "Interaction force estimation during manipulation of microparticles," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 950–956, Oct. 2012.