2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Planar batting under shape, pose, and impact uncertainty

Jiaxin L. Fu, Siddhartha S. Srinivasa, Nancy S. Pollard and Bart C. Nabbe

Abstract— **This paper explores the planning and control of a manipulation task accomplished in conditions of high uncertainty. Statistical techniques, like particle filters, provide a framework for expressing the uncertainty and partial observability of the real world and taking actions to reduce them. We explore a classic manipulation problem of planar batting, but with a new twist of shape, pose and impact uncertainty. We demonstrate a technique for characterizing and reducing this uncertainty using a particle filter coupled with a lookahead planner that maximizes information gain. We show that a two-step planner that first acts for information gain and then acts to maximize the expectation of achieving a desired goal is effective at managing shape, pose and impact uncertainty.**

I. Introduction

Real-world manipulation often must take place in conditions of high uncertainty. We may not know the exact shape of an object we are to manipulate, our sensors provide noisy information, and parameters of the world such as surface properties may not be exactly known.

Various strategies have been proposed for handling uncertainty in grasping and manipulation tasks.

One strategy is to take actions that reduce uncertainty given the availability of simple sensors (e.g., [1]) or no sensors at all (e.g., [2]). In such a framework, actions are akin to funnels that map a larger set of states into a smaller set, and a plan consists of a sequential composition of funnels that take all (or most) states to the goal [3], [4]. In some situations, open-loop strategies can produce periodic trajectories that are stable even in the presence of uncertainties (e.g., [5]). Note that uncertainty is modelled as a set, each point being equally important, and the plan guaranteed to take any point in this set to the goal.

Another strategy is to decouple the problem into two parts: an observer that monitors uncertainty and a controller which takes the best estimate of the observer and runs a deterministic strategy based on that. However, in most nonlinear systems, this is an artificially imposed decomposition and there is no guarantee that the controller's strategy will not interfere with the observer's ability to observe. Furthermore, the dynamics of manipulation are most often highly nonlinear and there have been a select few successful attempts at writing observers, for pushing [6] and for palmar manipulation [7].

An alternative strategy is to model the uncertainty as a probability mass, where points that are more likely to occur are given more weight, and treated preferentially, by the controller. Such statistical techniques allow for a quantification of uncertainty and the likelihood of a plan's success, as well as the construction of controllers and observers that can reason and trade-off between gaining information and achieving the goal. Statistical observers like Kalman filters and particle filters, statistical controllers based on solving Markov Decision Processes (MDPs), and many variations thereof have been extremely successful in solving a wide variety of problems in mobile robotics, like the simultaneous localization and mapping (SLAM) of a mobile robot in an unknown environment and localization of a target that is evading search [8]. We refer the reader to [9] for a detailed description and literature survey on probabilistic techniques.

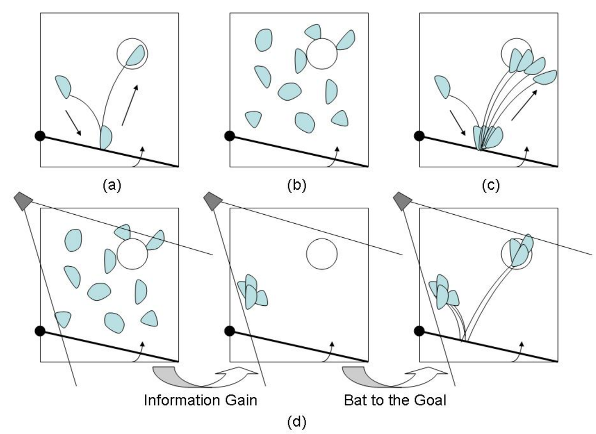

Fig. 1. Planar batting: a.Ideal situation; b.Pose and shape uncertainty; c.Impact noise;d.Two-step strategy: batting for information gain followed by batting to the goal

In this paper, we explore the use of statistical observation and control techniques to solve a version of a classic manipulation problem of planar batting with the added complexity of uncertainty in shape and pose. We observe that the manipulation problem possesses a unique structure that is unlike the mobile robotics problems for which these techniques have been successfully applied, thereby requiring the application of a unique strategy.

The goal of our planar batting problem is to bat an object into a desired goal state. The object moves under the action of gravity on a planar table, collides with walls, and is batted by a single degree of freedom revolute arm located at the bottom of the table (Fig.1). A number of researchers have addressed the problem of batting a spherical ball, in a variety of contexts like juggling, ping-pong [10], [11], 3D batting

Jiaxin Fu and Nancy Pollard are with The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA - 15213, USA jfu,nsp@ri.cmu.edu

Siddhartha Srinivasa and Bart Nabbe are with Intel Research Pittsburgh, 4720 Forbes Ave., Suite 410, Pittsburgh, PA - 15213, USA siddhartha.srinivasa, bart.c.nabbe@intel.com

[12]–[14], and devil-sticking [15].

We add two sources of uncertainty to the problem. First, the true shape of the object belongs to one of a finite number of shapes. Also, the true pose of the object is completely unknown, to begin with, as shown in Fig.1(b). Second, much like the real world, we model the uncertainty that is expected at impact. As a result, the uncertainty in shape and pose increases sharply after each impact.

To give the robot a fighting chance of actually completing the task, we add a planar pinhole camera to observe the object, albeit with noisy sensor measurements. The camera image is a segmented line composed of the object and the background.

A unique characteristic of our problem is that unlike the funnel framework, where actions collapse states and reduce uncertainty, in our problem, due to the noise during impact, actions expand states and *increase* uncertainty.

Another unique characteristic of our problem is that it is dynamic, with two distinct phases: a flight phase, where errors do not accumulate and we can zero in quickly on an estimate of object state based on camera readings, and a collision phase, where large errors are introduced, and our estimate of object state scatters.

Apart from the two-phase feature, the batting problem is different from other mobile robot problems that use particle filter in that the batting problem has only intermittent control instead of continuous control. Furthermore, one wrong action taken by the batter in the batting problem may lead to catastrophic consequences such as batting the object off the edge of the table.

These characteristics of our problem motivated a two-phase planner where we first take action to gain state information and then take a second action to achieve the goal given a model of uncertainty during the collision phase (Fig.1(d)). Of particular note, we observe (§VI) that considering the single best object state during the second action has a better performance than considering the entire distribution of particle states.

The rest of the paper is organized as follows. In §II and §III we describe the motion and impact models chosen for our problem. In §IV, we describe the implementation of a particle filter for tracking our belief of the shape and pose of the object. In §V, we describe our strategies for planning and observation to solve the planar batting problem. In §VI, we show the results of our strategies and discuss limitations and future work in §VII.

II. PROBLEM STATEMENT

This paper examines planar batting with object shape and pose uncertainty and impact uncertainty by a one joint robot arm in a gravitational field. We focus on finding batting control that can deal with the uncertainties and bat the object to the goal with the information from camera observation.

In our problem, we start with a known set of object shapes, but we do not know which instance we are batting. The object state space is then a 7D space and object states can be denoted as: Q=(q,$\dot{q}$,d)=($x,y,\varphi,\dot{x},\dot{y},\dot{\varphi},d$), where d represents the index of the shape instance in the object shape set, (x,y,φ) represents the object's position, and ($\dot{x},\dot{y},\dot{\varphi}$) represents the object's velocity.

The robot arm's state is (θ,$\dot{\theta}$), which are the joint angle and the angular velocity of the arm.

Observation information is available from a 2D camera, which has a 1D screen line. The observation information is the line segment on the screen line projected from the object.

We count the object as falling in the goal area, as long as the object's COM is within δ distance of the goal position, assuming the goal is an area instead of an exact point. The orientation of the object is not currently considered when evaluating whether it reaches the goal.

III. BATTING AND BOUNCING

We model the batting and the bouncing of the ball off the walls of the environment as collisions between rigid bodies with nonzero friction and restitution.

To compute the impact during the collisions, we use the collision model proposed by Chatterjee and Ruina [16]. The impact P relates the velocity of the object before and instantaneously after collision.

According to the model, given a pre-collision relative velocity of V_{bef} and the local mass matrix M at the contact point, the impulse P is a linear combination of two components: an impulse under a perfectly plastic and frictionless impact

$$P_I = -(\frac{n^T V_{\text{bef}}}{n^T M^{-1} n})n, \tag{1}$$

where n is the contact normal, and an impulse under the condition of perfectly plastic and perfectly sticking impact

$$P_{II} = -MV_{\text{bef}}, \tag{2}$$

Then a candidate impulse, $\overline{P}$ can be defined as:

$$\bar{P} = (1+e)P_I + (1+e_t)(P_{II} - P_I), \tag{3}$$

where 0≤ e ≤ 1 is the coefficient of restitution and -1≤e_t≤ 1 is the coefficient of tangential restitution.

If this candidate falls in the friction cone, then it is the final impulse resulting from the collision. Otherwise, we project it onto the friction cone surface, that is:

$$P = (1+e)P_I + k(P_{II} - P_I), \tag{4}$$

where

$$k = \begin{cases} 1+e_t, & \text{if } \| \bar{P} - nn^T P_{II} \| \leq \mu n^T \bar{P} \\ \frac{\mu(1+e)n^T P_I}{\|P_{II} - nn^T P_{II}\| - \mu n^T (P_{II} - P_I)}, & \text{otherwise} \end{cases} \tag{5}$$

where μ is the friction coefficient.

To account for the noise and perturbation during the impact, we include normally distributed noise about the direction of the computed impulse:

$$P' = P + \epsilon \tag{6}$$

IV. Shape and pose uncertainty

We use a particle filter to track the distribution of the shape and pose of the object through time. For detailed description of the particle filter technique, see [9]. Briefly, in this problem, each particle represents a state of the object. We start with a set of particles that are uniformly distributed in the state space, assuming no particular good guess of the state. Then at each time step, with the information of the current control and the current observation from the camera, the particle filter updates and resamples the particle set.

The particle set at time step t can be denoted as:

$$S_t = \{s_{t,1}, s_{t,2}, ..., s_{t,m}\} \tag{7}$$

where m is the number of particles in the set and each $s_{t,i}$ is a 7D vector, representing a possible state of the object.

At each time step, we update the particle set with the following steps:

1. Use physical rules described in the previous section to update the state of each particle in the set from the previous time step.

$$S_t = \text{PhysicalUpdate}(S_{t-1}). \tag{8}$$

For each particle, this can be either ballistic motion or collision with the batter or the walls of the environment, depending on the previous state of that particle.

2. For each particle, compute a the probability of the current camera observation given the current state:

$$w_{t,i} = P(o_t|s_{t,i}) \tag{9}$$

where o_t is the current camera reading.

The camera readings are represented by the two end points of the perspective projection of the object onto the screen line of a planar pinhole camera. To add realism, we add normally distributed noises to the camera readings.

3. Resample the particles with probability proportional to W_t=$w_{t,1}, w_{t,2}, ..., w_{t,m}$, which fills the particle set with the same number of particles.

We observe that the particle set converges towards the true object state during the ballistic phase of the true object and then diverges again due to the noise injected at impact.

V. Planning

A blind strategy could only bat open loop and not take into consideration any information feedback throughout the process. For instance, we can let the batter follow a preset trajectory, such as:

$$\theta = A \sin(\omega t) \tag{10}$$

where A is the magnitude and $-\pi/6 \leq \theta \leq \pi/6$.

We propose a two-step planer where the planner first finds the control that maximizes a measure of information gain for the particle filter, followed by batting to reach the goal.

A. Information gain

During the information-gain step, the batter does an n-timestep look ahead to choose one controller from a discrete set of open loop controllers that would yield the most converged particle set at time t, where $t \leq n$. The controllers are chosen from the set of cyclic controllers in Eqn.10. We bat the particle set using the selected controller to the time step t which has the lowest entropy. Approximating the (x, y) position of the particles by a 2D Gaussian model, we use the entropy of the distribution as a measure of its convergence:

$$H = \frac{1}{2}(1 + \log 2\pi + \log \det(\Sigma)) \tag{11}$$

where Σ is the covariance matrix of the particles. When the set is relatively converged, the entropy is low and we believe that the particles have a high probability of representing the true object state.

B. Batting to the goal

1) Open-loop batting with perfect information: The problem of successfully batting an object whose shape and pose are completely known is hard and interesting by itself. Like a trick shot in billiards, there can exist strategies that bounce the object off many walls. There can also exist strategies that require multiple bats to send the object to the goal. Finally, noise is a factor that must be considered in any strategy - for example, batting the object multiple times may only decrease the probability that it reaches the goal.

While the computation of an optimal policy given all of the above complexity is possible, and very interesting, we choose computational speed over optimality, primarily because we are interested in an algorithm that is near real-time so that it can be implemented on a real system, where the time between bats is small.

To achieve speed-up, we search only for the best single bat to the goal. We also discretize the control space (of bat angle and angular velocity at which the particle next hits the bat) into a finite set of possible actions (similar to techniques in [17]). We simulate the effect of all of the actions for the duration of one bat(*i.e.*, until the object hits the bat a second time; there is no restriction on the number of impacts between the object and the stationary walls) and pick the discretized action that minimizes a cost function f which is quadratic in the distance between the center of mass of the object and the center of the goal region. We then run an optimizer (MATLAB's `fminsearch`) to polish the solution.

In the presence of impact noise, however, there is a range of costs returned for the same action, since the trajectory, and thereby the cost, is affected by the noisy impact.

We would like to find a control $u \in \mathbb{U}$ that does the best, even in the presence of uncertainty in the objective function f induced by the uncertainty in the impact. We frame this as a problem of maximizing the probability of doing better than any other control v for a given f:

$$u^* = \arg\max_u \Pr\left[\forall v \in \mathbb{U}, f(u) \leq f(v)\right] \tag{12}$$

Because speed is of the essence, we use an approximate algorithm to solve Eqn.12. We sample a value for the noise and fix it. We compute the solution u_i^* to Eqn.12 for the fixed noise. We repeat this for n different values of the noise. From the n solutions $\{u_1^*, \ldots u_n^*\}$, we pick the one that is statistically the most representative. We observed that the u_i^* are tightly clustered and unimodal and pick the mean of this cluster as our representative control.

We experimented with two strategies for the objective function f, one minimizing the weighted sum of the distance between the particles and the goal, the other minimizing the distance between the most likely particle and the goal.

While both strategies maximize the expectation of success, there is no guarantee that either of them will actually succeed in a given trial.

2) Batting the whole set: In this strategy, we specify the objective function as minimizing the weighted sum of distance from the COM of every particle in the set to the goal.

3) Batting the best particle: Instead of batting the whole set, another strategy is to bat the particle which has the highest probability.

VI. Simulation results

To test the planning algorithms, we ran a set of simulation experiments in Matlab. To understand the value of different choices made in designing our algorithms, we test the following strategies:

1. Open loop. This is a straw man against which to compare all strategies which make use of information and planning. The bat simply performs the open loop trajectory of Eqn.10 with A=$\pi/6$ and $\omega = \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. These values of ω were chosen to provide a variety of controls without adding extreme amounts of energy to the objects. These are the same values of ω that are used in batting for information gain.

2. Closed loop, bat the entire particle set with no uncertainty in planning. Here we perform one bat for information gain and then find the optimal control to bat the entire particle set towards the goal.

3. Closed loop, bat the best particle with no uncertainty in planning. Here we perform one bat for information gain and then select the best particle. The best control is then found for this particle. We compare this strategy to 2) to determine whether choosing the best particle or batting all particles is a better choice.

4. Closed loop, bat the best particle with uncertainty in planning. This scenario is identical to 3), but uncertainty at impact is considered during planning. We compare this strategy to 3) to determine whether considering impact uncertainty during planning is a better choice.

5. Closed loop, bat the best particle with no uncertainty in planning and world. This scenario is identical to 3), except that impact uncertainty is also not simulated in the world. This scenario is unrealistic but serves as a point of comparison because no realistic scenario could ever improve upon this one.

A. Experiment setup

All the following simulation were run under a -1**m/**s^2 gravitational field (pointing downwards), with e=0.95, e_t=0.75 and μ=0.8. The size of the table and the positions of the camera and the batter are shown in Fig.3. The width of the camera screen is 2. The world frame origin is defined at the batter joint. The batter has a moving range between -π/6 and $\pi/6$. The red filled circle is the goal at (7,3) with a radius of 0.8; The green object is an example of the initial object state. There are seven smooth convex objects in the object set (Fig.2). The camera readings have a normally distributed noise with σ=0.1**m** and the impulse has a normally distributed noise of σ=0.0001**Kg·m/s**. We start with a set of particles that are uniformly distributed in the 7D state space, assuming no particular good guess of the start state and the initial number of particles is 200.

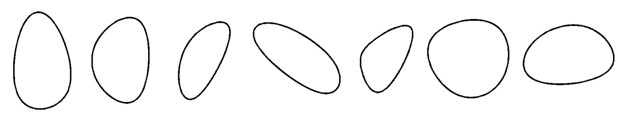

Fig. 2. Shapes used in the experiment.

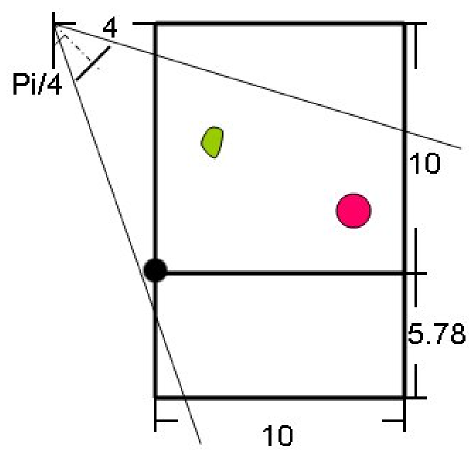

Fig. 3. Table, batter and camera setup

B. Particle filter

The particle filter uses the observation from the camera to resample the particle set, which discards the particles that have less probability of being the true object state and keeps multiple copies of the ones that are of high probability. Over time, the particles would converge toward the true state, but the uncertainty in the collision would then cause them to diverge. Fig.4 shows the entropy over time and some examples of the corresponding states of the particles. The converging phase happens when there are few collisions and good camera readings, while the diverging phase happens when there are many collisions and few good camera readings.

We identify the most likely particle at the end of the first step (the information gain step) and proceed with planning assuming this particle represents the true state of the object.

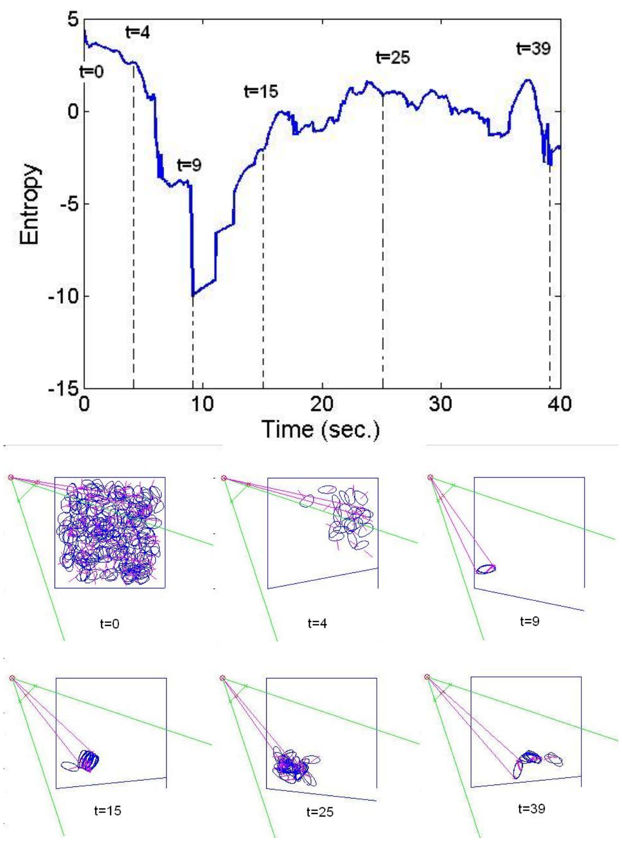

Fig. 4. Entropy over time and examples of the corresponding states of the particles

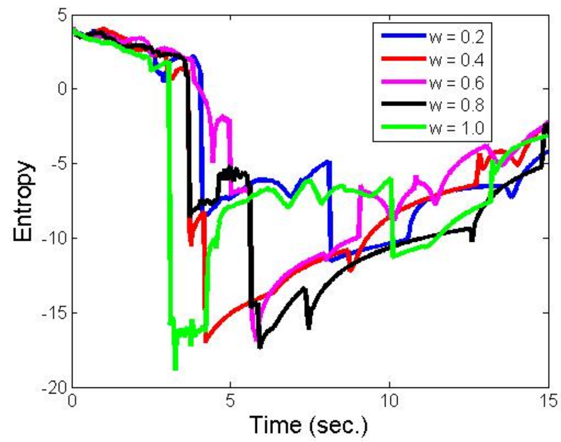

Fig. 5. Entropies of five control candidates during the first step.

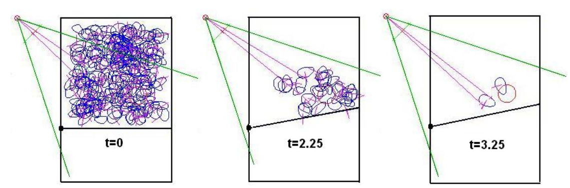

Fig. 6. Screen shots of the particles during the first step.

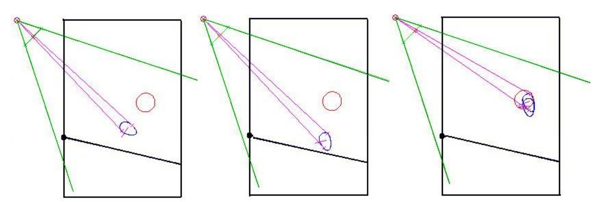

Fig. 7. Screen shots of the particles during the second step.

C. Two-step lookahead

For the closed-loop planner, in the first step we choose a control that results in the least entropy for a chosen amount of time. Fig.5 shows the entropies over time resulting from the five control candidates, respectively. The control represented by the green line (ω=1.0) will be chosen in this figure and the batter will use this control until the lowest point on the green line.

Fig.6 and Fig.7 show some screen shots from the simulation during the two steps. Note that in Fig.7 the particles start to diverge after the impact, which accounts for the more widely distributed particles in the right hand figure (after impact) vs. the center figure (just before impact).

D. Planning strategies

Table I shows the number of the times the object enters the goal under different control strategies for ten trials of different initial object states in two hundred seconds.

A. Iteratively do a t-timestep look ahead first for a control that leads to lowest entropy, then search for a control that leads to minimum weighted sum of distance between all the particles in the set and the goal.

B. Iteratively do a t-timestep look ahead first for a control that leads to lowest entropy, then search for an optimal control that bats the best particle to the goal. The planning part does not take into account the impact uncertainty.

C. Iteratively does a t-timestep look ahead first for a control that leads to lowest entropy, then search for an optimal control that bats the best particle to the goal. The

D-I. Openloop controls. The batter follows the following planning part takes into account the impact uncertainty. trajectory:

$$\theta = A \sin(\omega t) \tag{13}$$

where $\omega = \{0.2, 0.4, 0.6, 0.8, 1.0\}$, respectively for E-I and *A*=0.4. A has the most counts of reaching the goal. However this is under a world without noise in the second step of the two-step control strategy and with no uncertainty in planning. Although it is not realistic, it shows that the two-step control strategy works well when there is no noise. C is batting the best particle with no uncertainty in planning. It performs worse because when impact uncertainty is not taken into account in planning step and this impact uncertainty can cause the object's trajectory to depart from the best trajectory planned. However, when we plan with uncertainty, as in B, the performance improves significantly. It is still not as good as the result under a world without impact noise, but this is expected, as we can decrease the impact of the impact uncertainty on the result through careful planning, but we can never eliminate the uncertainty entirely. Batting the whole set does not work well, perhaps because it searches for a control that works well over the entire distribution of particles. The control has to compensate for the particles that are far away from the real object state and this requirement has an adverse effect on the good particles' performance. The open loop controls (E-J) have some interesting trends. For instance, E does much better than the other open loop controls, but they all do worse than the closed loop controls.

VII. Discussion

In summary, we present a two-step algorithm for planar batting under shape and pose uncertainty. Our algorithm first takes one step to gain information about object state and then executes a second step to perform a desired task–in this case, to bat the object toward the goal.

We handle shape and pose uncertainty simultaneously. However, in our experiments, shape uncertainty was very easily dismissed, because the state of the object collapsed to a single shape very rapidly. Incorrect shapes were not consistent with camera readings for very long, even given the expected uncertainty in camera readings.

It was more difficult to compensate for pose uncertainty. Our problem has the interesting property that noise is added to our estimate of object state through events which are discrete in nature and relatively far apart in time. In between these events, our sensors allow us to nearly localize an object, but each event once again increases uncertainty, making it difficult to plan many steps into the future.

We had two interesting findings while developing an algorithm to handle this type of pose uncertainty. First, as Table I illustrates, choosing an action based on our single best guess at particle state performs much better than considering the entire distribution of particle states. We are better off ignoring the distribution of likely particle states and optimizing based on our single best guess. For some intuition for why this may be true, consider that the best action for one particle may differ substantially from the best action for another particle, even if the states are similar. The mean of the two actions, then, may be good for neither particle, and so this mean action may result in a lower expected score than optimizing for either particle alone.

A second interesting finding, also supported by the results shown in Table I, is that when choosing an action, it was important to model uncertainty at impact. Simply considering the effect of the most likely impact response produced significantly worse results than considering the results over the distribution of impact responses. In this case, the distribution of uncertainty could not be ignored and was important for good performance. Our intuition here is that the single best guess for particle state before impact is likely to be correct or nearly correct, yet the particle state after impact will over many trials reflect the distribution of states captured by our impact uncertainty model.

In terms of future work, our immediate plan is to construct a robot platform to test whether our two-step batting strategy works as well in practice as in simulation. We would also like to identify good strategies in a more automatic fashion. For this paper, we had to specify that the robot would first perform a single action for information gain followed by a second action to accomplish the task. We expect to pursue two parallel directions for identifying good strategies, the first based on performing a more comprehensive search through the space of possibilities and the second based on learning from human demonstration. We also believe that particle filter approach and two phase strategy can be applied to a broader range of manipulation tasks, for example, to address uncertainties caused by impact as the hand acquires an object and manipulates it into the palm for use in a particular task.

VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of Dhruv Batra, Geoffrey Gordon, Matthew Mason, Ali Rahimi, Rahul Sukthankar, and Garth Zeglin. Jiaxin Fu is partially supported by the Intel Summer Research Fellowship and by NSF awards CCF-0343161, IIS-0326322, CNS-0423546, and ECS-0325383.

References

[1] D. Kang and K. Goldberg, "Sorting parts by random grasping," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 146–152, 1995.

[2] M. Erdmann and M. Mason, "An exploration of sensorless manipulation," in *Int. Journal of Rob. Res.*, vol. 4, 1988, pp. 369–378.

[3] M. Mason, "The mechanics of manipulation," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1985.

[4] R. Burridge, A. Rizzi, and D. Koditschek, "Sequential composition of dynamically dexterous robot behaviors," *International Journal of Robotics Research*, vol. 18, pp. 534–555, 1998.

[5] R. Ronsse, P. Lefevre, and R. Sepulchre, "Open-loop stabilization of 2d impact juggling," in *6th IFAC Symposium on Nonlinear Control Systems*, Stuggart, Germany, 2004, pp. 1157–1162.

[6] Y.-B. Jia and M. Erdmann, "Pose and motion from contact," *International Journal of Robotics Research*, vol. 18, no. 5, pp. 466–490, 1999.

	δ	0.07	0.08	0.09	0.10	0.11	0.12	0.13	0.14	0.15
Close loop	A.Bat best particle, No uncertainty in planning and world	19	22	26	33	39	44	50	55	58
	B.Bat best particle, with uncertainty in planning	**11**	**15**	**21**	**25**	**29**	**32**	**37**	**45**	**50**
	C.Bat best particle, no uncertainty in planning	10	11	14	18	21	25	30	32	35
	D.Bat whole set	4	5	5	10	10	13	15	18	23
Open loop	E.ω=0.2	8	12	16	16	16	18	18	20	22
	F.ω=0.4	6	6	6	8	8	10	11	11	12
	G.ω=0.6	5	5	7	7	7	11	13	14	14
	H.ω=0.8	4	5	6	7	10	13	14	15	15
	I.ω=1.0	3	4	6	9	9	13	14	18	19
	J.ω=0	0	0	0	0	0	1	1	1	1

TABLE I

NUMBER OF TIME PASSING THE GOAL IN 200 SECONDS. THE NUMBERS LISTED ARE THE SUM OF 10 TRIALS OF DIFFERENT INITIAL OBJECT STATE. THE OBJECT IS CONSIDERED AS FALLING IN THE GOAL WHEN THE CENTER OF MASS OF THE OBJECT IS WITHIN δ DISTANCE TO THE GOAL ;

[7] M. Moll and M. Erdmann, "Reconstructing the shape and motion of unknown objects with active tactile sensors," in *Algorithmic Foundations of Robotics V*, J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, Eds. Springer Verlag, 2003, pp. 293–310.

[8] S. T. Brian Gerkey and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *International Journal of Robotics Research*, vol. 25, no. 4, p. 299, 2006.

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

[10] R. Andersson, "Understanding and applying a robot ping-pong player's expert controller," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1989, pp. 1284–1289.

[11] D. C. Bentivegna, A. Ude, C. G. Atkeson, and G. Cheng, "Learning to act from observation and practice," *International Journal of Humanoid Robotics*, vol. 1, no. 4, 2004.

[12] E. W. Aboaf, S. M. Drucker, and C. G. Atkeson., "Task-level robot learning: Juggling a tennis ball more accurately," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1989, pp. 1290–1295.

[13] M. Buhler and D. E. Koditschek, "From stable to chaotic juggling: Theory, simulation, and experiments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1990, pp. 1976–1981.

[14] A. A. Rizzi and D. E. Koditschek, "Progress in spatial robot juggling," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1992, pp. 1976–1981.

[15] S. Schaal, C. G. Atkeson, and S. Botros., "What should be learned?" in *Proceedings Seventh Yale Workshop on Adaptive and Learning Systems*, 1992, pp. 199–204.

[16] A. Chatterjee and A. Ruina, "A new algebraic rigid body collision law based on impulse space onsiderations," in *ASME Journal of Applied Mechanics*, vol. 65, no. 4, 1998, pp. 939–951.

[17] C. Black and K. Lynch, "Planning and control for planar batting and hopping," in *36th Annual Allerton Conference on Communication, Control, and Computing*, 1998, pp. 724–733.