Planning for Dynamics under Uncertainty

Dicong Qiu
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
dq@cs.cmu.edu

Karsh Tharyani
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ktharyan@cs.cmu.edu

Abstract— **Planning under uncertainty is a frequently encountered problem. Noisy observation is a typical situation that introduces uncertainty. Such a problem can be formulated as a Partially Observable Markov Decision Process (POMDP). However, solving a POMDP is nontrivial and can be computationally expensive in continuous state, action, observation and latent state space. Through this work, we consider a restricted POMDP problem, where we alleviate the dependency of the latent state on the controllable action. Our proposed approach involves using an Extended Kalman Filter (EKF) for latent state estimation and a Particle Filter for goal estimation in conjunction with an iterative Linear Quadratic Regulator (iLQR) to find an optimal trajectory that minimizes the cumulative model cost. As a means to evaluate the feasibility and optimality of our solution, we compare this approach against the naive strategy of planning to only the goal rolled out from the immediate observation made by the agent.**

I. Introduction

Uncertainty is a frequently encountered and, at most of the time, an inevitable problem especially in robotic planning tasks, which can be formulated as a Partially Observable Markov Decision Process. we consider a restricted POMDP problem in continuous state, action, observation and latent state space, where we alleviate the dependency of the latent state on the controllable action. Though this simplification assumption reduce the complexity of the problem, it is still highly nontrivial. The continuity of the problem space, the uncertainty in observation and the highly nonlinear dynamics jointly complicate the problem, making it intractable with traditionally search-based planners. So we proposed solving this problem with an Extended Kalman Filter for latent state estimation and a Particle Filter for goal estimation in conjunction with an iLQR solver to find an optimal trajectory towards the best estimate of the goal.

We demonstrate our proposed approach in a toy problem against the naive strategy of planning to only the goal rolled out from the immediate observation made by the agent, where the objective is plan an optimal action sequence for a robot with an arm to catch a balloon that flies in a chaotic trajectory. The problem involves a 4-DOF robot (with a 2-DOF mobile base and a 2-DOF manipulator mounted on the base) and a flying balloon. However, the sensors on the robot are noisy, and hence, the robot cannot observe the ground-truth state of the balloon. Instead, the robot may acquire noisy observation of the balloon. We also assume that the state of the robot is fully observable. As is in most cases, the control input to the robot will be the joint and the base accelerations, which thus results in a continuous dynamical system of the robot. We formally formulated the aforementioned problem in the POMDP framework. Since the state and action of the robot does not affect the balloon dynamics, the problem is a simplified (restricted) POMDP. Furthermore, we assume that the robot has prior knowledge of the dynamics of the balloon. And this assumption is valid, because in practice the dynamics of a balloon can be learned through prior observations in a self-supervised manner.

II. Related Work

People have been investigating the problem of planning under uncertainty for many decades. The witness algorithm mentioned in [1] proposes to prune *t*-step policy trees by considering dominance relationships. It provides a lucid explanation of the formulation of Belief MDPs from the POMDPs and provides a Q-function formulation to prune the *t*-step trees. Another work in this area is the Point Based Value Iteration (PBVI) algorithm [2], which introduces an approximated solution to Belief-Markov Decision Process (Belief MDP). PBVI interleaves the value backup iterations (similar to Real Time Dynamic Programming) with the expansion of the belief set. Hence, at each step the the convex hull corresponding to the optimal policy is approximated with a set of belief points and the value associated with them.

III. Problem Formulation

We consider the problem of restricted partially observable Markov decision process (POMDP), with a simplification assumption that the latent state $\mathbf{z} \in \mathbb{R}^Z$ does not depend on the controllable action $\mathbf{u} \in \mathbb{R}^U$ or the fully observable state $\mathbf{x} \in \mathbb{R}^X$. It therefore infers the latent state is also independent from the dynamics f involving the fully observable state and the controllable action. In our formulation of the problem, the the dynamical system consists of two independent dynamics, the agent dynamics f and the environmental dynamics g.

$$\mathbf{x}^{(t+1)} = f\left(\mathbf{x}^{(t)}, \mathbf{u}^{(t)}\right) \qquad \mathbf{z}^{(t+1)} = g\left(\mathbf{z}^{(t)} \middle| \phi\right)$$

where the latent state transition depends on the environmental prior ϕ. The agent has no access to the latent state, but it may infer (estimate) the latent state through observation $\mathbf{o} \in \mathbb{R}^O$.

$$\mathbf{o}^{(t)} \sim O\left(\mathbf{o} \middle| \mathbf{z}^{(t)}, \mathbf{x}^{(t)}\right) \doteq \mathbb{P}\left(\mathbf{o} \middle| \mathbf{z}^{(t)}, \mathbf{x}^{(t)}\right)$$

where O is the observation model that gives the probability distribution of the immediate observation $\mathbf{o}$ conditioned on the latent state $\mathbf{z}$ and the fully observable state $\mathbf{x}$. The objective is to find an optimal control sequence $\mathbf{U}^*$ that minimizes the cumulative model cost J.

$$J(\mathbf{U}) = \sum_{t=0}^{T-1} \mathcal{L}\left(\mathbf{x}^{(\mathbf{t})}, \mathbf{u}^{(\mathbf{t})} \middle| \mathbf{z}^{(\mathbf{t})}, M\right) + \mathcal{L}_f\left(\mathbf{z}^{(\mathbf{T})}, \mathbf{x}^{(\mathbf{T})} \middle| M\right)$$

$$\mathbf{U}^* = \arg\min_{\mathbf{U}} J(\mathbf{U})$$

where $\mathbf{U} = \left\{\mathbf{u}^{(t)}\right\}_{t=0}^{T-1}$ is the sequence of control throughout the planning horizon T, M is the given environmental information, and $\mathcal{L}$ and $\mathcal{L}_f$ are respectively the running loss and the final loss.

IV. Method

In order to solve the problem, there are several fundamental aspects to concern. Firstly, due to the uncertainty of the latent state, the variables in the problem domain are not fully observable. In order to maintain a best estimate $\hat{\mathbf{z}}^{(t)}$ along with uncertainty $\hat{\mathbf{\Sigma}}^{(t)}$ of the latent state $\mathbf{z}^{(t)}$ at any time t from the observation history $\mathbf{O} = \left\{\mathbf{o}^{(t)}\right\}_{t=0}^{T-1}$, we adopt the Extended Kalman Filter [3], due to the highly nonlinear dynamics of the environment. Secondly, the both the fully observable state and the latent state are not stationary, so it is necessary to forward simulate the dynamics in order to find the goal state. In the case where the cumulative model cost J does not depends on the intermediate latent states but the final one $\mathbf{z}^{(T)}$, we use a forward particle filter [4] to simulate the best estimate $\hat{\mathbf{z}}$ latent state forward up till time T, to estimate the goal state $\hat{\mathbf{z}}^{(T)} \sim \mathbf{PF}\left(\mathbf{z} \middle| g, \hat{\mathbf{z}}^{(t)}, \hat{\mathbf{\Sigma}}^{(t)}, T\right)$. Such an approach becomes feasible since we assume that the environmental dynamics g does not depends on the controllable action $\mathbf{u}$ or the fully observable state $\mathbf{x}$. Jointly, the two steps mentioned above resemble a forward simulating version of the Unscented Kalman Filter (UKF) [5]. Finally, we employ an iLQR solver [6] to plan towards the goal state.

V. Experiment

In our experiment, we apply our proposed approach to the balloon catching problem in 2D (see: Appendix I), as shown in figure 4. The holonomic robot with a 2-DOF manipulator makes an 8-dimensional fully observable state $\mathbf{x}$ (of position and velocity), that is deterministically controllable (by mobile base and joint accelerations) by the control input $\mathbf{u}$. Whereas the state of the balloon (of position and velocity) is not directly observable, which is the latent state $\mathbf{z}$ in our framework. And we assume that the observation of the balloon state lies in the same space as the latent state, but with uncertainty that conforms to a Gaussian distribution $\mathbf{o}^{(t)} \sim \mathcal{N}\left(\mathbf{z}^{(t)}, \Sigma\left(\mathbf{x}^{(t)}, \mathbf{z}^{(t)}\right)\right)$. The balloon is moving in chaotic trajectories, which makes the environmental dynamics highly nonlinear (see: Appendix II). We set up the running loss $\mathcal{L}$ to avoid the obstacles while introducing the minimal energy, and the final loss $\mathcal{L}_f$ drives the end-effector of the robot to reach the best estimate $\hat{\mathbf{z}}$ of the latent state.

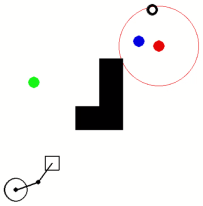


Fig. 1: A shot from the video clip, where the hollow square at the end of the link represents the end-effector, the hollow circle surrounding the balloon is the uncertainty scale, the black hollow ring represents the instantaneous observation, the blue dot is the estimated state of the balloon, and the green dot is the best estimate of the goal.

$$\mathcal{L}\left(\hat{\mathbf{z}}, \mathbf{x}, \mathbf{u} | M\right) = \sum_{\text{obs}=1}^{N_{\text{obs}}} \left\|\mathbf{p}_R - \mathbf{p}_{\text{obs}}\right\|_2^2 + \lambda \left\|\mathbf{u}\right\|_2^2$$

$$\mathcal{L}_f\left(\hat{\mathbf{z}}, \mathbf{x}, \mathbf{u} | M\right) = \left\|\hat{\mathbf{z}} - \mathbf{FK}_E\left(\mathbf{x}\right)\right\|_2^2$$

where $\mathbf{FK}_E\left(\mathbf{x}\right)$ is the nonlinear forward kinematics of the end-effector that maps the state of the robot to the state of the end-effector, M is the map information, $\mathbf{p}_R = \begin{bmatrix} x_R & y_R \end{bmatrix}^\mathsf{T}$ is the location of the robot base and $\mathbf{p}_{\text{obs}} = \begin{bmatrix} x_{\text{obs}} & y_{\text{obs}} \end{bmatrix}^\mathsf{T}$ is the location of the obstacle obs.

Table III summarizes the results of our experiment, where we compare our approach (EKF+PF) against a naive strategy baseline (without EKF/PF).

Ours (With EKF+PF)		Baseline (Without EKF/PF)	
$\bar{\epsilon}$	Succ' Rate	$\bar{\epsilon}$	Succ' Rate
0.305 ± 0.262	80.00%	1.085 ± 0.231	0.00%

TABLE I: Summary Results

where $\bar{\epsilon}$ is average distance (in meters) between the end-effector and balloon in the last time step. We observed an 80.00% success rate of catching the target with our solution strategy, as compared to the naive strategy. We also observe that the final cost (distance of the end-effector from the balloon) is fairly low in both the naive strategy and our strategy.

VI. Conclusion

As can be inferred from the experiment results, both the average final distance error $\bar{\epsilon}$ and the success rate indicates that the method (with EKF) we proposed outperformed the baseline (without EKF). It suggests that our proposed method by handling the uncertainty with Extended Kalman Filter and particle filter performs better in the our POMDP set up, compared to the naive strategy that assumes the instant observation reflects the best estimate of the latent state.

References

[1] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

[2] Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pages 1025–1032, 2003.

[3] Gerald L Smith, Stanley F Schmidt, and Leonard A McGee. *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. National Aeronautics and Space Administration, 1962.

[4] Pierre Del Moral. Non-linear filtering: interacting particle resolution. *Markov processes and related fields*, 2(4):555–581, 1996.

[5] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[6] Yuval Tassa, Nicolas Mansard, and Emo Todorov. Control-limited differential dynamic programming. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1168–1175. IEEE, 2014.

Appendix I
Balloon Catching Environment

To test the proposed method, we consider solving the problem of catching a moving target under partial observation, such as a simulated flying balloon that move stochastically. In the balloon catching environment, as shown below in Figure 2, the state and action spaces are continuous, and we consider it as a discrete time system.

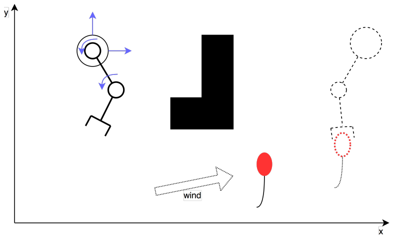


Fig. 2: An illustration of the balloon catching problem.

A. Robot Behavior

The state of the robot is deterministic, where we consider the position of the mobile base of the robot and the joint angles of the robot arm. Since we assume that the mobile base can move holonomically, so it is not necessary to model the orientation of the mobile base. As for the controllable action of the robot, which includes the behaviors of the mobile base and the robot arm, we assume that we can adjust the linear acceleration of the mobile base and the angular acceleration of the joints of the robot arm. It is a valid assumption that we may directly control the acceleration, since in physical robotic systems, we control the robots through electric voltage and current, which eventually results in the change of acceleration. If we assume a perfect model of how the change of electric voltage and current will result in the change of acceleration, we can then control the acceleration directly.

$$\mathbf{x} = \begin{bmatrix} x_R & y_R & \theta_1 & \theta_2 & \dot{x}_R & \dot{y}_R & \dot{\theta}_1 & \dot{\theta}_2 \end{bmatrix}^\mathsf{T}$$
$$\mathbf{u} = \begin{bmatrix} \ddot{x}_R & \ddot{y}_R & \ddot{\theta}_1 & \ddot{\theta}_2 \end{bmatrix}^\mathsf{T}$$

where x_R and y_R are the position of the robot mobile base, θ_1 and θ_2 are the joint angles, $\dot{x}_R$, $\dot{y}_R$, $\dot{\theta}_1$ and $\dot{\theta}_2$ are the corresponding velocities, and $\ddot{x}_R$, $\ddot{y}_R$, $\ddot{\theta}_1$ and $\ddot{\theta}_2$ are the corresponding accelerations. We also consider the state transformation of the robot, including its arm, is a linear transformation.

$$\mathbf{x}^{(t+1)} = \mathbf{A}\mathbf{x}^{(t)} + \mathbf{B}\mathbf{u}^{(t)}$$

More specifically,

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \Delta t & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 \\ 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & \Delta t \end{bmatrix}$$

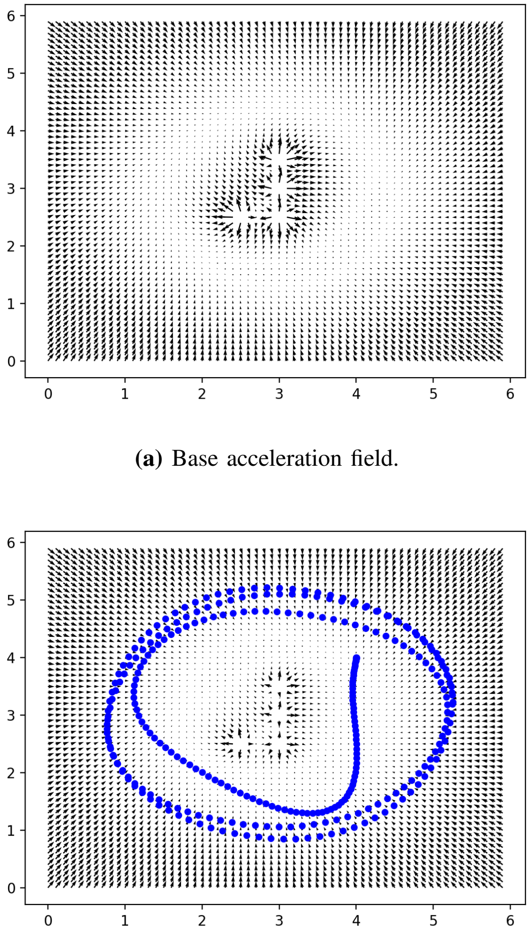
where Δt is the time interval of a simulation step.

B. Balloon Behavior

The balloon moves in a deterministic manner following some rules. In order to make the problem more realistic, we also introduce the environmental prior, which is some prior information about the environment that could affect the behavior of the balloon. Formally, the balloon moves in accord to a dynamic model conditioned on an environmental prior.

$$\mathbf{z}^{(t+1)} = g\left(\mathbf{z}^{(t)} \middle| \phi\right)$$

where $\mathbf{z} = \begin{bmatrix} x_T & y_T & \dot{x}_T & \dot{y}_T \end{bmatrix}^\mathsf{T}$ is the state of the balloon that is not directly observable to the robot, ϕ is the environmental prior, and $g(\cdot)$ is the dynamic model of the balloon. In order to simulate the behavior of the balloon, i.e. defining g concretely, we consider the superposition of multiple vector fields that simulates the wind along with a damping mechanism to prevent the balloon from moving too fast. These vector fields include: (1) a circular field, which makes the balloon rotate about the center, (2) a centralization field, which pulls the balloon towards the center, and (3) multiple dispersion fields, which drive the balloon away from the obstacles. Together, these fields determine the base acceleration of the balloon at any location. The environmental prior in this particular example is a scalar that is positively related to the speed of the wind, or says, the strength of the circular field. The damping mechanism is implemented by introducing a friction that is positively related to the instant velocity of the balloon. The detailed formulation of balloon behavior can be found in Appendix II. And a base acceleration



(a) Base acceleration field.

(b) Example trajectory.

Fig. 3: Visualization of an example base acceleration field for the balloon and an example balloon trajectory superposed on the base acceleration field, where the center is $(3.0, 3.0)$, there are 4 obstacles in the center area, and the example balloon trajectory is generated under $\phi = -1.0$ from a randomly selected starting location.

field example and a balloon trajectory example are shown in Figure 3.

Though the trajectory of the balloon is deterministic given its starting state $\mathbf{z}^{(0)}$ and the environment prior ϕ, the ground-truth state of the balloon cannot be directly observed by the robot. Instead, noisy observation of the balloon is given to the robot according to some observation model, a example of which can be one that introduces more noise (higher uncertainty) when the robot is farther away from the balloon. Additionally, the noisy environmental prior given to the robot at the beginning, which will also introduce uncertainty to the balloon state estimation. An intuition of the environmental prior uncertainty is that the robot may observe the environment property directly since the robot and the target are in the same environment, but not exactly since there is always noise in the observation. For instance, the robot is able to feel the wind in the environment that will also influence the balloon, but the robot is not able to feel the exact wind speed without any observation error.

Appendix II
Balloon Behavior Formulation

The balloon moves following a deterministic trajectory if its initial state $\mathbf{z}^{(0)}$ and the environmental prior ϕ is given. In order to simulate the wind that is characterized by the environmental prior ϕ and influences the trajectory of the balloon, we consider a superposition of three different kinds of fields, namely (1) a circular field that makes the balloon rotate about the center, (2) a centralization field that pulls the balloon towards the center, and (3) multiple dispersion fields which drive the balloon away from the obstacles. As given in the following, are concrete definitions of the unit circular field, the unit centralization field, and the unit dispersion field, assuming the origin $(0, 0)$ as center.

$$F_{\text{circular}}\left(x, y\right) = \frac{1}{\sqrt{x^2+y^2}} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$F_{\text{central}}\left(x, y\right) = \begin{bmatrix} -x \\ -y \end{bmatrix}$$

$$F_{\text{disperse}}\left(x, y\right) = \frac{1}{x^2+y^2} \begin{bmatrix} x \\ y \end{bmatrix}$$

Together, these vector fields simulate the influence of the wind applied on the balloon. In other words, they determines the base acceleration a_o of the balloon at any state $\mathbf{z}$ in the scene caused by the environmental factor, i.e. the wind. However, the velocity balloon will increase without limit if there is no constraints on it. In order to constrain the velocity of the balloon, we also introduce an additional accelerate a_f caused by friction, which has an opposite direction to the instant velocity of the balloon and is proportional to the second order of the balloon velocity scale, thus to suppress the growth of the velocity of the balloon. The joint acceleration a of the balloon is then a sum of the base acceleration and the acceleration cased by friction.

$$a_0\left(\mathbf{z}\middle|\phi, \mathbf{c}, \{\mathbf{o}_i\}_i^k\right) = \begin{bmatrix} \lambda_{\text{circular}} \\ \lambda_{\text{central}} \\ \lambda_{\text{obstacles}} \end{bmatrix}^\mathsf{T} \mathbf{F}$$

$$\mathbf{F} = \begin{bmatrix} F_{\text{circular}}\left(x_T - x_C, y_T - y_C\right) \\ F_{\text{central}}\left(x_T - x_C, y_T - y_C\right) \\ \sum_{i=1}^k F_{\text{disperse}}\left(x_T - x_{O_i}, y_T - y_{O_i}\right) \end{bmatrix}$$

$$a_f\left(\mathbf{z}|\mu\right) = -\mu\left(\dot{x}_T^2 + \dot{y}_T^2\right) \begin{bmatrix} \dot{x}_T\left(\sqrt{\dot{x}_T^2}\right)^{-1} \\ \dot{y}_T\left(\sqrt{\dot{y}_T^2}\right)^{-1} \end{bmatrix}$$

$$a\left(\mathbf{z}\middle|\phi, \mu, \mathbf{c}, \{\mathbf{o}_i\}_i^k\right) = a_0\left(\mathbf{z}\middle|\phi, \mathbf{c}, \{\mathbf{o}_i\}_i^k\right) + a_f\left(\mathbf{z}|\mu\right)$$

where $\mathbf{z} = \left[x_T, y_T, \dot{x}_T, \dot{y}_T\right]^\mathsf{T}$ is the instant state of the balloon, $\mathbf{c} = \left[x_C, y_C\right]^\mathsf{T}$ is a predefined center, $\{\mathbf{o}_i\}_i^k$ is a set of obstacle locations with $\mathbf{o}_i = \left[x_{O_i}, y_{O_i}\right]^\mathsf{T}, \forall i \in \{1, 2, \cdots, k\}$, μ is the friction factor as a constant scalar, and

$\lambda_{\text{circular}}$, λ_{central} and $\lambda_{\text{obstacles}}$ are the weights determining the contribution of different fields to the base acceleration of the balloon.

Appendix III
Experiment Details

A. Parameters

Below are the parameters we use for our experiment.

- The base of the robot is circular with a radius of 25cm. The lengths of the links are 50cm each.
- The balloon is of radius 12.5cm.
- The balloon is assumed caught when the robot's end-effector is within 30cm from the balloon.
- The observations of the balloon are drawn from a Gaussian distribution with a standard deviation 20% of the distance between the base of the robot and the position of the balloon. The noise radius, thus, decreases when the robot is nearer to the balloon than it previously was, or increases as their relative distance increases.

B. Observations and Results

We performed a total of 10 trials in each of the two cases: 1) When the robot tries to only catch the target based on its observations (without EKF), and 2) When the robot uses a state estimator to predict the state of the balloon (with EKF), and then tries to catch it. At the end of each trial, we noted if the robot was able to catch the balloon, and what was the distance of the end-effector from the balloon after the control at the final time-step had been executed. Table II shows our observations from these trials, and Table III summarizes our results.

Experiment	With EKF		Without EKF	
Trial	ϵ	Caught	ϵ	Caught
1	0.411	NO	1.13	NO
2	0.254	YES	0.968	NO
3	0.258	YES	0.863	NO
4	0.25419	YES	1.052	NO
5	0.116	YES	0.947	NO
6	0.181	YES	1.704	NO
7	0.128	YES	1.013	NO
8	1.05	NO	1.25	NO
9	0.130	YES	0.920	NO
10	0.270	YES	0.998	NO

TABLE II: Observations

where ϵ is distance (in meters) between the end-effector and balloon in the last time step,

Ours (With EKF+PF)		Baseline (Without EKF/PF)	
$\bar{\epsilon}$	Succ' Rate	$\bar{\epsilon}$	Succ' Rate
0.305 ± 0.262	80.00%	1.085 ± 0.231	0.00%

TABLE III: Summary Results

where $\bar{\epsilon}$ is average distance (in meters) between the end-effector and balloon in the last time step.

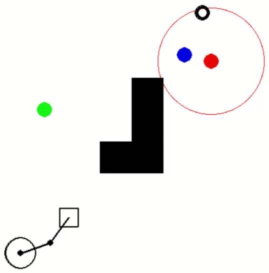


Fig. 4: A shot from the video clip

C. Video Clip

The video clip of the planner with the state-estimator is available[1]. An image from the clip is shown below in Figure 4. The robot is shown in the bottom-left of the clip. It indicates the base and the 2-DOF arm of the robot. The hollow square at the end of the link is the end-effector. The red dot is the balloon. The hollow circle surrounding the balloon is the uncertainty associated with the location of the balloon in the 2D world. An observation (the black hollow dot or ring) is drawn from a gaussian distribution with the standard deviation the size of the uncertainty ball (one standard deviation) around the balloon. The blue dot is the estimated state of the balloon from the Extended Kalman Filter (EKF). It must be noted that the state of the balloon also comprises of the velocity of the balloon, however, it is not depicted in the clip. The green dot is the fowarded state of the particles filtered around the state estimate. Hence, the green dot is the target state to which the robot plans in the planning horizon. The blocked rectangles indicate the obstacles in the environment.

[1] `https://www.youtube.com/watch?v=fVyPd2h9JKQ`