Flower pollination using point-cloud measurements and POMDP

Bernardo Martinez R. Junior

Abstract— **Flower pollination using robotic systems is a problem that involves planning and control under a lot of uncertainty, since obtaining a correct description of a flower in space as it involves several steps: flower detection, segmentation, and relative pose estimation. To make decisions under this uncertainty, in this paper, the problem was modeled as a partially observable Markov decision process (POMDP) with an observation model that uses a normal distribution with both mean and covariance values dependent on the state. To quantify the dependency of the mean and covariance, a simulated environment was setup with a single flower and point cloud data was collected from several relative poses from a discretized state space. The pose was estimated from the centroid and surface normal of the point-cloud and compared to the actual relative pose. After building the models, the problem was solved using SARSOP. The final policy was able to pollinate the flower correctly in 20% of the trials.**

I. Introduction

Pollinators are responsible for the reproduction of many plants by transferring pollen from one flower to another. This "free service" provided by nature is vital for fruit and seed production, which are the basis for human agriculture. Nevertheless, over the last decades, the population of natural pollinators has experienced a big decline caused by several reasons, as exposure to pesticides, loss of habitat, the spread of diseases, and climate change [1].

Although the survival of the natural pollinators is essential, and their loss would lead to extreme economic impact and other important externalities [2], precision robotic systems can help to execute the pollination tasks in a cost-effective and stable form. Most of the methods explored so far do not make the robot interact with a flower as a bee would, by touch. One exception is the Bramblebee robot that consists of a robotic arm mounted on top of a mobile base and that touches the flowers using its end-effector [3].

One of the challenges of designing a robot that interacts with a flower is to achieve a good estimate of the flower pose before proceeding with the pollination procedure. In their previous work, this estimate was achieved by executing a predefined sequence of actions with the robotic arm [4]. However, this process can be performed with a higher degree of autonomy using reinforcement learning. Considering that the flower localization problem is subject to a lot of uncertainty caused by factors like sensor noise, sensor resolution, flower unknown geometry, and flower detection and segmentation, one possible solution framework is to use partially observable Markov decision processes (POMDPs) to allow for principled decision making under conditions of uncertain sensing [5].

The author is with the Department of Mechanical and Aerospace of the Benjamin M. Statler College of Engineering and Mineral Resources at West Virginia University, Morgantown, WV, USA. Email: bm00002@mix.wvu.edu

The motivation behind this work is to compute a policy to control a robotic pollinator under the uncertainty of the sensor measurements. The sensor provided is a stereo camera, which is capable of generating a point cloud of the environment containing the flower the robot needs to pollinate. The approach will be divided into three parts. The first part consists of generating a simulation framework that uses a stereo camera to get point cloud data from flower models. This point cloud will be segmented to extract a single flower and segmented clouds will be recorded for a range of possible configurations of the flower with respect to the camera frame. The second part consists of creating the observation model given by $h(s,a)$ and $R_t(s)$ for the POMDP solver. The measurement function $h(s)$ will be created using the center of mass of the point cloud to obtain the position of the flower and using Principal Component Analysis (PCA) to extract a normal vector of the point cloud and, consequently, its orientation, assuming it looks like a disk. The noise function $R_t(s)$ is expected to be also dependent on the states because seeing the flower from more oblique orientations will assume the flower disk model fails. To get this function we want to use the true pose and check the deviation from the estimated values.

$$\max_{s,a} \sum_{t=0}^{H} \gamma^t \; r(s,a) \tag{1}$$

The third part consists of solving the POMDP problem, shown in 1, and obtaining a policy that takes as inputs the belief state of the flower b_t and outputs the best action, by maximizing the expected reward, which will take into account reward successful pollination actions, penalize actions moving the arm to reduce uncertainty, and strongly penalize unsuccessful pollination actions. The POMDP problem will be solved using the offline approximated solution method SARSOP [6], which takes as inputs all possible state combinations that we expect to encounter and outputs best actions for each of them. This policy is the expected outcome of this work.

The rest of this paper is structured as follows: the method for estimating the flower pose from the images coming from a stereo camera is shown in Section II. The details of the POMDP problem are given in Section III. The solution is evaluated in Section IV. Section V compiles the conclusions and proposals for future work.

II. Flower Pose Estimation

When a robotic system pollinating is to correctly detect a target flower in the image coming from its cameras, and to estimate this flower's position and orientation with respect to the robot. If it was possible to solve these steps with very high accuracy, the problem of pollinating would reduce to a simple motion planning problem for the robotic arm and mobile base. However, these steps usually provide a very uncertain result. In this section, we describe our approach on how to estimate the flower position and orientation.

In run-time the agent is provided the point-cloud of the flower, a set X of N data points, which are converted to a single observation of its pose. The pose estimation problem is divided into two parts: estimating the position and estimating the orientation of the flower, as shown in Fig. 1. The position estimation can be easily performed by just taking the centroid $\hat{\mathbf{x}} = [x, y, z]^T$ of the collection of points, $\hat{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i$, where $\mathbf{x} \in X \subset \mathbb{R}^3$.

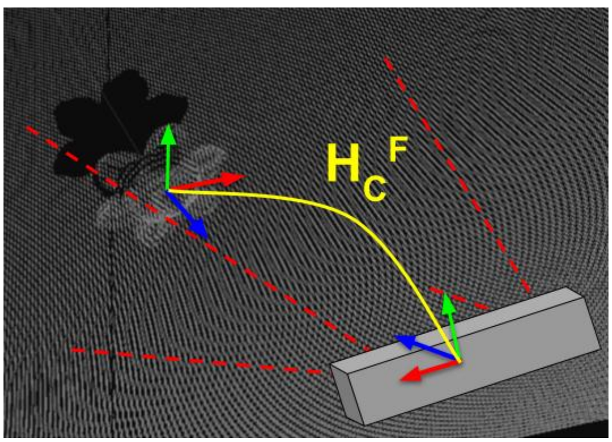


Fig. 1: Flower pose estimation. The goal is to obtain the homogeneous transformation matrix H_C^F from the camera frame C to the flower frame F.

The orientation of the flower can be approximated as the normal of the plane tangent to the disk. Obtaining the plane tangent to the disk, in its turn becomes the least-square plane fitting estimation problem. The normals can be computed via the Principal Component Analysis (PCA) by calculating the covariance matrix $\mathcal{C} \in \mathbb{R}^{3\times 3}$ of the point cloud and analyzing the eigenvalues and eigenvectors of this matrix [7]. The covariance matrix and the eigenvalue problem are given by:

$$\mathcal{C} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \hat{\mathbf{x}}) \cdot (\mathbf{x}_i - \hat{\mathbf{x}})^T, \ \mathcal{C}\mathbf{v} = \lambda \mathbf{v} \tag{2}$$

The eigenvectors resulting from this problem form an orthogonal frame, the principal components, and the eigenvector corresponding to the smallest eigenvalue is the approximation of the normal $\pm\hat{\mathbf{n}} = [n_x, n_y, n_z]^T$. However, there is an ambiguity in this calculation, where the normal can be pointing to one side or the other of the plane. To solve the problem, one can use the knowledge of the viewpoint $\mathbf{v}_P$, enforcing $\hat{\mathbf{n}} \cdot (\mathbf{v}_P - \hat{\mathbf{x}}) > 0$. In our problem, as the flower is centered at the image of the camera ($x \approx 0$, $y \approx 0$) and the camera is located at the origin of its coordinate frame ($\mathbf{v}_P = [0, 0, 0]^T$), this condition translates to having a negative n_z.

III. POMDP

A. Background

Partially observable Markov decision processes (POMDPs) is a generalization of the MDP framework, where the agent cannot directly measure its state. The POMDP model can be specified as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, T, Z, R, \gamma)$. The set of states $\mathcal{S}$ contains all possible states s representating the relative pose between the end-effector and the target flower, this pose contains both position and orientation. The set of actions $\mathcal{A}$ contains all actions that take the end-effector from state s and moves it to state s' . The transition between states is given by the function $T(s, a, s') = p(s'|s, a)$, the probability of reaching state s' when the action a is taken from state s. At each time step, the agent makes an observation o. This observation is modeled with a probability function $Z(s, a, o) = p(o|s, a)$. The reward function $R(s, a)$ is a signal represented by real number that encodes the outcome of taking an action a from state s. The goal of the agent is to maximize the expected cumulative reward by choosing a sequence of optimal actions.

B. Choosing a solver

SARSOP is a point-based algorithm that computes approximate solution to POMDP problems [6]. It uses *value iteration* starting from a initial policy, uses upper and lower bounds for the value and performs backup operations until the value function V converges. To deal with one of the main problems of reinforcement learning, *the curse of dimensionality*, many predecessors of the SARSOP used sampling a set of points from the belief space $\mathcal{B}$ instead of dealing with the whole space. Later the concept of reachable space $\mathcal{R}(b_0)$ was introduced to further reduce the order of the problem by sampling only the portion of space that can be reached from a initial belief b_0. SARSOP extends this idea by pruning the space, removing the suboptimal actions, i.e. points were to be sampled only in the optimally reachable space $\mathcal{R}^*(b_0)$.

C. The pollination problem

The POMDP problem explored in this work can be described as finding a policy that returns the optimal actions given a current belief of the relative pose of the flower with respect to the pollinator robotic arm.

1) The state space: The state space is given by the end-effect position in the global coordinates, the flower position in global coordinates, the flower orientation in global coordinates, and whether the pollinator was actuated or not. These states fully define the problem and also provide a link between different states and actions, since the end-effect can move with negligible uncertainty. The state-space was discretized as shown in Fig. 2: the end-effector position was discretized in a 5x6 grid, with 0.3m step size (30 possibilities). The flower was allowed to be in subset of the end-effector grid (a 3x3 grid) (9 possibilities). Further, the flower was allowed to have 5 possible orientations (5 possibilities), in 45 degrees steps. Finally, there are 2 possibilities for the actuated/not-actuated state. This adds up to $30 \times 9 \times 5 \times 2 = 2700$ states.

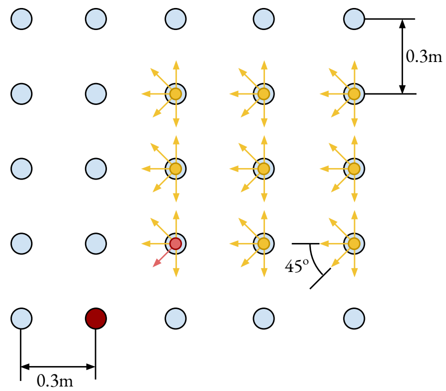


Fig. 2: State-space. The possible positions of the end-effector are shown by the blue dots. The positions of the flower are shown by the yellow dots. The possible flower orientations are shown by the yellow arrows. In red a configuration where a pollination actuation would be successful is shown.

2) The action space: After discretizing the space, the action space contains one action for each discrete coordinate of the end-effector (30 actions). Additionally, we include the action of moving to pollinate the flower (1 action).

3) The transition model: The end-effector is capable of moving with respect to its coordinate frame with a negligible uncertainty. Given a initial state $s = (p_{e1}, P_f, \text{not actuated})$, if the corresponding motion action is selected, the state is transitioned to $s' = (p_{e2}, P_f, \text{not actuated})$ with probability one. If the pollinate action is selected, the state is transitioned to $s' = (p, P_f, \text{actuated})$ with probability one.

4) The observation model: The observation probabilistic model is given by the function $p(o|s,a) \sim \mathcal{N}(h(s,a), R_t(s))$. In this problem we chose to simulate a system with a single flower and a camera positioned at several relative poses using ROS and Gazebo. A bramble flower model and a RealSense D435i plugin were used. For that, we used spherical coordinates to discretize the world, as it was done for the state-space. The radial coordinate r was discretized in steps of 0.15m, the spherical angles were discretized in steps of $15°$. Then for each possible combination of these three parameters, point-cloud data was collected for a short period of time. The motivation behind that is to empirically obtain metrics for the covariance of the position and orientation estimation. From this covariance, for each relative positioning, we calculate the probability of observing different orientations of the flower and that is used as the observation model.

5) Reward function: A negative value of -10 is given at each non-successful time step where the agent chooses to change the relative pose. A large negative value of -100 is given if the agent tries to pollinate and fails. A large positive value of +100 is given if the agent tries to pollinate and succeeds.

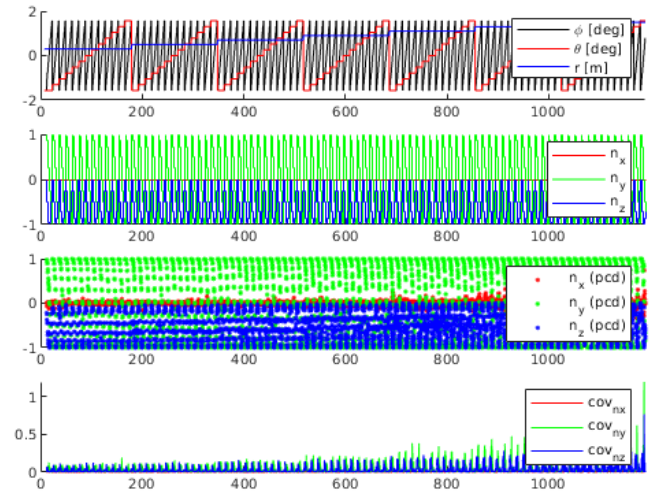


Fig. 3: Simulation data and normal estimation. From the top, the first graph shows the discretization of space in simulation. The second shows the true flower normal. The third shows the expected flower normal. The last graph is obtained from the error squared between the true and estimated flower normal.

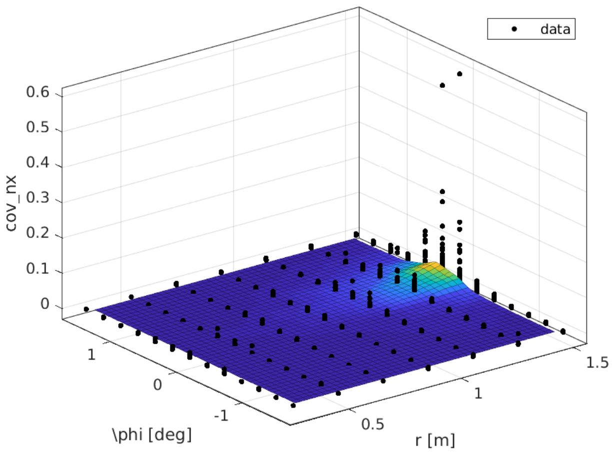
TABLE I: Simulation results for 1000 runs.

Number of Simulations	Expected Total Reward	Number of Simulations	Expected Total Reward
100	-79.59	600	-73.90
200	-75.08	700	-74.08
300	-73.86	800	-73.29
400	-72.01	900	-72.36
500	-74.56	1000	-72.33

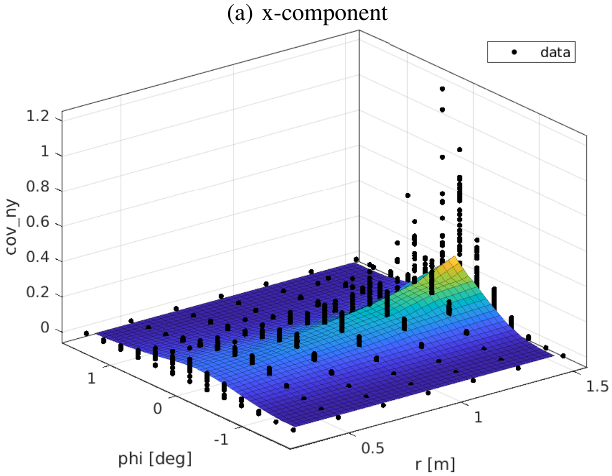
IV. Results

The models were loaded into SARSOP in 7.98s. The upper bound was initialized as -100 and the upper bound was initialized as -2.32. After 291 trials and 77.13s of computation time, the bounds converged to -70.3535, with precision of 6e-13. The number of backups was 2587, the number of Alphas was 65 and the number of beliefs was 136. To confirm the performance, the policy was simulated in 1000 runs. The results are shown in Table I. The Expected Total Reward obtained was -72.3247. The 95% confidence interval for the Exp. Total Reward was (-76.8148, -67.8347).

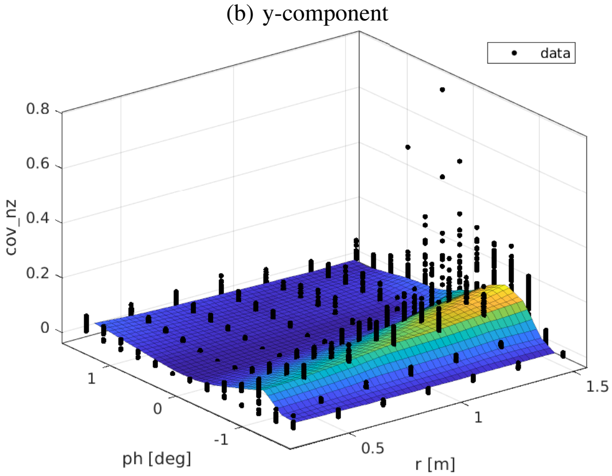
The solution, shown in Fig. 5 shows that the agent tries at least one step (and not more than two) in the direction it believes the flower, getting the reward of -10, and tries to pollinate. Once it is there, it tries to pollinate. One every five times it gets it right and receives +100. The remaining four times, it fails and receives -100. In average, it will take one step before trying to pollinate. This means, that in average each five runs it will get: $4 \times (-10 - 100) + 1 \times (-10 + 100))/5 = -350/5 = -70$. This means that the observation model is not correct, and the agent is not gathering enough extra information by approaching the flower.



(a) x-component



(b) y-component



(c) z-component

Fig. 4: Co-variance the x, y, z components in the normal estimation for the simulated data and linear interpolation. Only r and ϕ are considered since θ has negligible effect due to flower symmetry.

V. Conclusions and Future Work

The POMDP approach to solve robotic task is becoming more practical as it is able to solve problems with a greater range of states. The SARSOP solver was able to find an optimal policy for this pollination problem that had 2700 and 31 actions. The total expected reward obtained was approximately -70, which was confirmed by running a test with 10 batches of 100 runs. The policy obtained is not still satisfactory, since it is only successful at one in five runs. Nevertheless, the observation model does not seem to provide the correct information and a better implementation is desired. Further, the reward can be reshape to decrease the penalty of the end-effector staying in the same position and taking more observations.

References

[1] S. Kluser and P. Peduzzi, “Global pollinator decline: a literature review,” 2007.

[2] N. Gallai, J.-M. Salles, J. Settele, and B. E. Vaissière, “Economic valuation of the vulnerability of world agriculture confronted with pollinator decline,” *Ecological economics*, vol. 68, no. 3, pp. 810–821, 2009.

[3] N. Ohi, K. Lassak, R. Watson, J. Strader, Y. Du, C. Yang, G. Hedrick, J. Nguyen, S. Harper, D. Reynolds *et al.*, “Design of an autonomous precision pollination robot,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7711–7718.

[4] J. Strader, J. Nguyen, C. Tatsch, Y. Du, K. Lassak, B. Buzzo, R. Watson, H. Cerbone, N. Ohi, C. Yang *et al.*, “Flower interaction subsystem for a precision pollination robot,” *arXiv preprint arXiv:1906.09294*, 2019.

[5] M. T. Spaan, “Partially observable markov decision processes,” in *Reinforcement Learning*. Springer, 2012, pp. 387–414.

[6] H. Kurniawati, D. Hsu, and W. S. Lee, “SARSOP: Efficient point-based pomdp planning by approximating optimally reachable belief spaces.” Citeseer.

[7] R. B. Rusu, “Semantic 3d object maps for everyday manipulation in human living environments,” *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

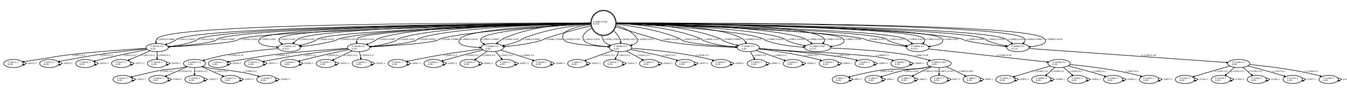


Fig. 5: Policy tree. The root node represents the state with maximum belief and the action selected in it. Each arrow represents a possible observation. The agent progresses in the tree until a final state is reached (after trying to pollinate the state just transitions to itself).