Optimal Control of a Planar Drone for Location of Radioactive Sources

Long Kiu Chung[1], Yixian Li[2], and Megan Elizabeth Ochalek[3]

[1]edgarclk@stanford.edu
[2]s031820@stanford.edu
[3]mochalek@stanford.edu

June 6, 2021

Abstract

Radioactive source localisation and retrieval via unmanned aerial vehicles (UAV) mounted with radiation detectors is an active area of research due to its ability to mitigate potential risks of human exposure to radioactive materials. In this project, we commanded and controlled a quadrotor drone to locate and retrieve a radioactive source in a two-dimensional (2D) field, utilizing a source location estimation algorithm previously developed by one of our team members. We first applied optimal control methods at each time step for a clear field and compared the results with that of a proportional-derivative (PD) controller. Additionally, we investigated flight scenarios with obstacles involved; this added layers of complexity both in sensing and trajectory planning.

Project Video: `https://www.youtube.com/watch?v=ps3MgFMJBfc`

1 Introduction

Nuclear engineers are frequently tasked with retrieving lost radioactive sources in an open environment. Luckily, with the rise of drone technology, it becomes feasible for a quadrotor drone to locate and proceed to the source to perform the removal, minimizing exposure to personnel involved.

However, previous work on the source location estimation algorithm did not take control or drone dynamics into account. In this project, we combine optimal control with the estimation algorithm for more realistic simulations and applications, optimizing the drone's trajectory to successfully perform the task on a 2D plane with and without non-attenuating obstacles.

The remainder of this document is organized as follows. In section 2 we review relevant literature and detail the source location estimation algorithm and control model to be used in our simulations. In section 3 we summarize our goal in one concise problem statement. In section 4 we present our proposed method. In section 5 we present our results for source localization and retrieval with and without non-attenuating obstacles. Finally, in section 6 we provide concluding remarks and discuss future directions.

2 Related Work

2.1 Source Location Estimation

Hazardous navigation and radioactive source retrieval is an active research field among the Health Physics community. Several trajectory generation schemes have been proposed, with methods ranging from Monte Carlo simulations, recursive Bayesian estimation, or even image reconstruction techniques [1–5]. Our project is based on a two-dimensional (2D) version of the algorithm developed for the open-space case by Chung, a member of our team, in [6].

In [6], the 2D problem was set up for a drone mounted only with location and radiation sensors, searching for a randomly generated radioactive source within an open field. At each time-step t_i, which is limited by the updating frequency of the radiation detector, the drone returns its location x_i, y_i, and the detector reading of dose rate, $\dot{D}_i$. Assuming the radioactive source follows the inverse-square law, the algorithm solves for the following problem using the Levenberg-Marquardt (LM) method at t_n:

$$\underset{k_{est}, x_{est}, y_{est}}{\text{argmin}} \sum_{i=0}^{n} (\dot{D}_i - \frac{k_{est}}{(x_i - x_{est})^2 + (y_i - y_{est})^2}) \tag{1}$$

where k_{est} is the guessed proportional constant and (x_{est}, y_{est}) is the guessed location of the source, which is used to update the drone's goal. The whole process repeats until the estimation location converges and is within reach, or if an iteration limit has been reached.

2.2 Drone Control

Knowledge of the drone's dynamics is critical for the application of control algorithms. For a 2D realization of a planar quadrotor, the system of dynamics is given by Wu and Sreenath in [7] with a slight fix on the sign:

$$m\ddot{x} = -F \sin\theta \tag{2}$$

$$m\ddot{y} = F \cos\theta - mg \tag{3}$$

$$J\ddot{\theta} = -M \tag{4}$$

where g is the acceleration of gravity; m, J are the mass and inertia of the planar quadrotor; F, M are inputs, representing thrust and moment exerted by the propellers. The dynamics from the above equation could be written as:

$$\dot{s} = f(s) + g(s)u + constant \tag{5}$$

where s is the stack of state variables as $[x\ y\ \theta\ \dot{x}\ \dot{y}\ \dot{\theta}]^T \in \mathbb{R}^6$, and u is the stack of input variables as $[F\ M]^T \in \mathbb{R}^2$.

3 Problem Statement

For this project, we designed a controller for trajectory optimization of a quadrotor drone retrieving radioactive sources. In order to simplify the drone dynamics, we considered the 2D planar case, such that the drone's flight path is constrained to a vertical plane. We first utilized iLQR to initially optimize a trajectory to the source with no obstacles present, then factored in obstacles that must be avoided. Finally, we compared our optimal control solution and trajectories to a proportional-derivative (PD) controller as a baseline.

4 Approach

4.1 Source identification with measurement

For the open-space case and the non-attenuating obstacles case, at each t_i where a measurement is taken, the source location estimation can simply be updated using (1), whose performance and failure modes were already thoroughly analyzed in [6].

If the obstacles attenuate the reading, we can redefine the optimization problem for p circular obstacles, assuming the location and size of all obstacles were known:

$$\underset{k_{est},x_{est},y_{est},\mu_{est,1},\cdots,\mu_{est,p}}{\operatorname{argmin}} \sum_{i=0}^{n}(\dot{D}_i - \frac{k_{est}}{(x_i - x_{est})^2 + (y_i - y_{est})^2} \prod_{j=1}^{p} e^{-\mu_{est,j} d_j(x_i,y_i,x_{est},y_{est})}) \tag{6}$$

where $\mu_{est,j}$ is the estimated linear attenuation coefficient for the j^{th} obstacle and $d_j(x_i, y_i, x_{est}, y_{est})$ is the length of intersection between the j^{th} obstacle and the straight line from (x_i, y_i) to (x_{est}, y_{est}), which can be calculated by solving:

$$d_j(x_i, y_i, x_{est}, y_{est}) = \sqrt{(x_{int1,j} - x_{int2,j})^2 + (y_{int1,j} - y_{int2,j})^2} \tag{7}$$

$$(x_{int,j} - x_{c,j})^2 + (y_{int,j} - y_{c,j})^2 = r_j^2 \tag{8}$$

$$\frac{y_{int,j} - y_i}{x_{int,j} - x_i} = \frac{y_{est} - y_i}{x_{est} - x_i} \tag{9}$$

where $x_{int1,j}$ and $x_{int2,j}$ are the two real roots of $x_{int,j}$, $y_{int1,j}$ and $y_{int2,j}$ are the two roots of $y_{int,j}$, r_j is the radius of the obstacle, and $(x_{c,j}, y_{c,j})$ is the center of the obstacle. If solutions for $x_{int,j}$ and $y_{int,j}$ are imaginary, $d_j(x_i, y_i, x_{est}, y_{est}) = 0$.

We can further simplify the problem by assuming all μ_j (related to material compositions of the obstacles) are known, reducing (6) to:

$$\underset{k_{est},x_{est},y_{est}}{\operatorname{argmin}} \sum_{i=0}^{n}(\dot{D}_i - \frac{k_{est}}{(x_i - x_{est})^2 + (y_i - y_{est})^2} \prod_{j=1}^{p} e^{-\mu_j d_j(x_i,y_i,x_{est},y_{est})}) \tag{10}$$

That said, even with the simplification, the solution still does not converge to the correct location every time and is highly sensitive to simulation parameters due to the high non-linearity of the problem. We leave finding a better solver as future work and instead assume all obstacles are non-attenuating in subsequent simulations.

4.2 Open space control with constant update on source location

4.2.1 Algorithm structure

The overall algorithm structure is given in Algorithm 1 below.

Algorithm 1: Open space control with constant update on source location

```
Initialize $s_{cur}$ = [0 0 0 0 0 0], $x_{cur}$ = [0 0]
for $i \leq maxiter$ do
    Store previous $\mathbf{x}_g$ as $\mathbf{x}_{gp}$
    Read current sensor data from $\mathbf{x}_{cur}$ and update guessed source location $\mathbf{x}_g$ and set $s_g$ = [$\mathbf{x}_g$ 0 0 0 0]
    if $DIST(\mathbf{x}_{cur}, \mathbf{x}_g) \leq \epsilon$ and $DIST(\mathbf{x}_g, \mathbf{x}_{gp}) \leq \delta$ then
        Break
    Run PD($s_{cur}$, $s_g$) or iLQR($s_{cur}$, $s_g$) and get control sequence $u_{star}$
    Simulate the system forward with $u_{star}$ for one second, update $\mathbf{x}_{cur}$ and $s_{cur}$
* $\mathbf{x}$ denotes the stacked vector of [x y]
```

The algorithm iteratively performs the following: we take a current measurement, guess the source location, run the standard PD or iLQR based on our guess. We then simulate our system with some time steps (for our problem, the duration is one second) by applying the control policy from PD or iLQR, and we repeat the process until we meet the condition which gives us confidence in reaching the goal position.

4.2.2 PD control

We decided to first implement the PD control for the model described in section 2.2. This was done by deriving the following control laws for our state variables and then implemented via `Simulink`, as described in [8] at 8:03 in the video.

$$F = m(g + \ddot{y}_{des} + k_{v,z} * (\dot{y}_{des} - \dot{y}) + k_{p,z} * (y_{des} - y)) \tag{11}$$

$$M = k_{v,\theta} * (\dot{\theta}_c - \dot{\theta}) + k_{p,\theta} * (\theta_c - \theta) \tag{12}$$

$$\theta_c = -\frac{1}{g}(\ddot{x}_{des} + k_{v,x} * (\dot{x}_{des} - \dot{x}) + k_{p,x} * (x_{des} - x)) \tag{13}$$

4.2.3 iLQR

We will be reapplying the iLQR method at each time step t_i and define the cost function and discretized dynamics.

The iLQR algorithm utilizes the cost function that is defined by:

$$\frac{1}{2}(s_N - s_g)^T Q_f (s_N - s_g) + \frac{1}{2}\sum_{k=0}^{N-1}((s_k - s_g)^T Q (s_k - s_g) + u_k^T R u_k) \tag{14}$$

where s_N is the final state, s_k are the intermediate states, s_g is the target pose (defined as $[x_{des}\ y_{des}\ 0\ 0\ 0\ 0]$, where (x_{des}, y_{des}) is provided by (1)), u_k is the intermediate control, Q_f is the final state cost matrix, Q is the intermediate state cost matrix, and R is the control cost matrix.

To satisfy the performance, we would frequently adjust the different cost matrices, time step, and simulation time in problem setup.

4.3 Incorporating obstacle avoidance

To have the iLQR controller avoid non-attenuating obstacles, we calculate the collision factor $\alpha_{j,k}$ for the j^{th} obstacle at operating point $(\bar{s}_k, \bar{u}_k)$ in each iLQR iteration:

$$\alpha_{j,k} = r_j^2 - (\bar{s}_k' - \begin{bmatrix} x_{c,j} & y_{c,j} & 0 & 0 & 0 & 0 \end{bmatrix}) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} (\bar{s}_k - \begin{bmatrix} x_{c,j} \\ y_{c,j} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}) \tag{15}$$

If $\alpha_{j,k} > 0$, then we add $0.5\gamma\alpha_{j,k}$ to the stage-wise cost, where γ is a scaling factor.

5 Experiments

5.1 PD Control

For our baseline comparison for our optimal control, we implemented a PD controller on our quadrotor drone system, as described in (11) (12) and (13). The implementation is detailed in Algorithm 1. The PD controller is implemented for a time step of one second between each measurements. The result is shown in Fig. 1.

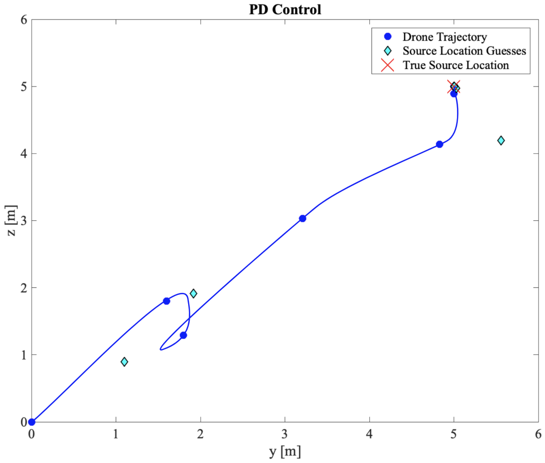

Figure 1: Trajectory of drone that begins at (0, 0) and uses proportional-derivative control to locate the source at (5,5).

5.2 Optimal Control – iterative Linear Quadratic Regulator (iLQR)

5.2.1 Open Space

We simulated two cases, where we used different state cost matrices. In particular, for the first case, we defined our state cost matrix $Q = diag[10, 10, 10, 2, 2, 2]$ and final state cost matrix $Q_f = 100*I_6$; for the second case, $Q = diag[3, 3, 3, 0.6, 0.6, 0.6]$, and $Q_f = 30 * I_6$. For both cases, we enforce $T = 5$, $dt = 0.2$, and control cost matrix $R = 0.1 * I_2$. The trajectory and control results are presented in Fig. 2 and Fig. 3.

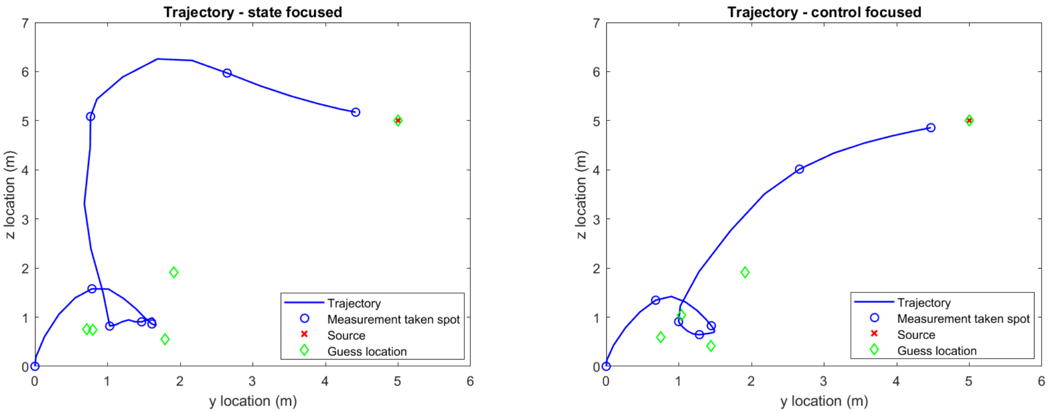

Figure 2: Trajectories of drone that begins at (0, 0) and uses iLQR control to iteratively locate the source at (5,5). The case on the left used a bigger state cost matrix.

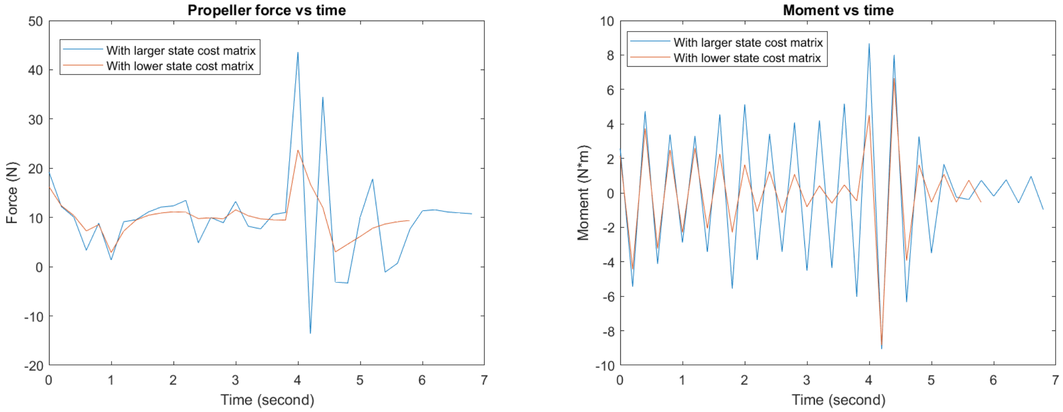

Figure 3: Control comparison on force and moment between cases with different cost matrices using iLQR.

From the two trajectory plots in Fig. 2, we see that, with higher cost on states, the drone navigated straight up and straight right. While it steadily rose and proceeded to the right if a lower cost on states is specified. From the comparison of the control input plots in Fig. 3, we see that, with lower cost on state, which gives more importance on control, the force and moment inputs were reduced. It is also interesting to see that with higher cost on state, the drone did not reach the goal position faster as we have expected. This is probably because we are constantly changing the guessed goal positions, and if the drone moves too fast, it may end up reaching the wrong source location early and need to rework to the correct source location.

5.2.2 Obstacle Avoidance

With the same cost matrices (the case with larger state cost) defined in 5.2.1 and based on the rationale stated in 4.3, we additionally defined the scaling constant $\gamma = 10000$, and an obstacle that is located at [1, 5] with a radius of 0.5. We obtained the performance results in Fig. 4 and Fig. 5.

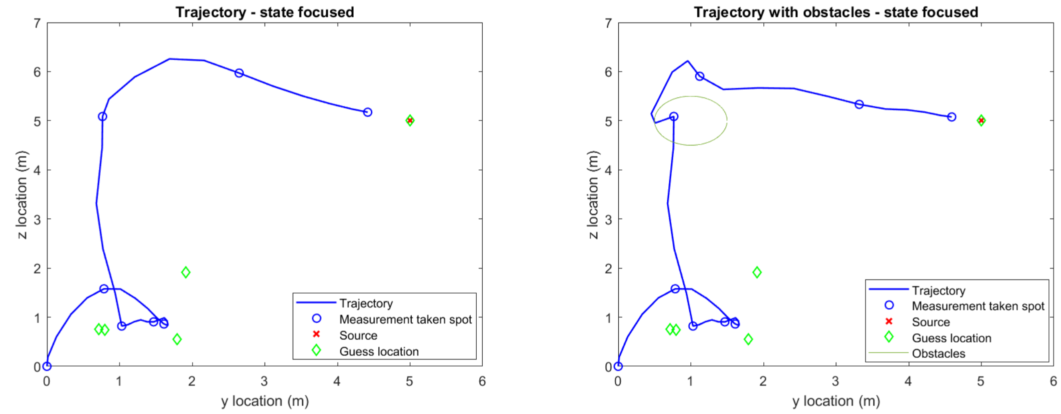

Figure 4: Trajectory comparison between open space case and obstacle case using iLQR. The obstacle case has a danger region specified with an ellipse.

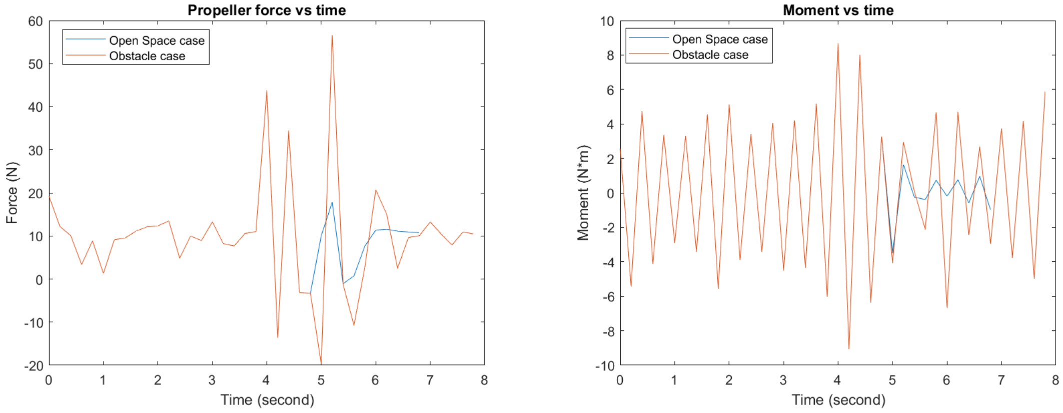

Figure 5: Control comparison on force and moment between open space case and obstacle case using iLQR.

From Fig. 4, we see that the drone plans to avoid the obstacle when it is within the danger region. From Fig. 5, we see that, when the drone encounters the obstacle, the time it takes is longer and the input is larger towards the end of the simulation.

Still, for the obstacle case, the algorithm and parameter need to be more carefully modified and tuned to yield a trajectory that circumvents the danger region entirely.

6 Conclusions and Future Work

With PD and iLQR controller, we managed to obtain feasible trajectories in an open space plan-and-simulate problem setting. In particular, we observed that: while PD control gives us a trajectory that tends to minimize its length, iLQR is more flexible and gives us different trajectories depending on how fast we want our drone be, and what control we want the drone to maintain. However, since we are utilizing soft constraints, the parameter tuning becomes a more difficult thing to accomplish. Additionally, we are using the controller with updating guess locations, this results in uncertainties as we simulate.

As stated, the iLQR controller was not able to avoid the obstacle very well, since the extra cost term does not guarantee enforcement. As future work, we wish to improve obstacle avoidance via convex optimization or model predictive control (MPC). For attenuating obstacles such as lead and steel, we would want to explore better solvers for the optimization problem in (6) or (10).

For real-life applications, we can also extend our experimentation on more complicated 3D quadrotor models, as well as using AirSim for visualization.

References

[1] Jerry Towler, Bryan Krawiec, and Kevin Kochersberger. Radiation mapping in post-disaster environments using an autonomous helicopter. *Remote Sensing*, 4(7):1995–2015, 2012.

[2] Jared A. Cook, Ralph C. Smith, Jason M. Hite, Razvan Stefanescu, and John Mattingly. Application and evaluation of surrogate models for radiation source search. *Algorithms*, 12(12), 2019.

[3] Shuangyue Zhang, Ruirui Liu, and Tianyu Zhao. Mapping radiation distribution on ground based on the measurement using an unmanned aerial vehicle. *Journal of Environmental Radioactivity*, 193-194:44–56, 2018.

[4] Ruirui Liu, Kathryn Higley, and Xinzhi Liu. Discussion on one algorithm for mapping the radiation distribution on contaminated ground. *Health physics*, 109(1):25—34, July 2015.

[5] Tairan Liu, Angela Di Fulvio, Long Kiu Chung, and Kimberlee J Kearfott. Radiation mapping for an unmanned aerial vehicle: Development and simulated testing of algorithms for source mapping and navigation path generation. *Health physics*, 120(3):321—338, March 2021.

[6] Long Kiu Chung, Andrew J. E. Kent, Margaret A. Cooney, Jordan D. Noey, Kyle J. Liebler, and Kimberlee J. Kearfott. Simulations and experimental verifications of an algorithm for radiation source mapping and navigational path generation. *Health Physics*, Publish Ahead of Print, 2021.

[7] Guofan Wu and Koushil Sreenath. Safety-critical control of a planar quadrotor. In *2016 American Control Conference (ACC)*, pages 2252–2258, 2016.

[8] Vijay Kumar. 2-d quadrotor control - planning and control.