Quadrotor Trajectory Optimization Using Potential Fluid Flows

*Final Project for MEAM5170: Control and Optimization with Applications in Robotics, taught by Professor Michael Posa in Fall 2022.

Nathaniel Ruhl
School of Engineering and Applied Sciences
University of Pennsylvania
Philadelphia, Pennsylvania
nruhl@seas.upenn.edu

Alec Lanter
School of Engineering and Applied Sciences
University of Pennsylvania
Philadelphia, Pennsylvania
alanter@seas.upenn.edu

Jason Li
School of Engineering and Applied Sciences
University of Pennsylvania
Philadelphia, Pennsylvania
lijason1@seas.upenn.edu

***Abstract*—Velocity potential functions can be applied to path planning by representing start positions, end goals, and obstacles as sources, sinks, and vortices. In this work, we use potential field theory, inspired by two-dimensional potential flows in fluid dynamics, to construct an artificial potential field that represents an obstacle-laden environment. We then compute a streamline which serves as a warm start initial guess for iLQR trajectory optimization. We task a differentially flat 2D-quadrotor with navigating procedurally-generated, random obstacle fields and compare the results between the novel warm start iLQR and unassisted cold start iLQR. Our analysis shows that warm start iLQR converges to a trajectory quicker and more reliably than cold start iLQR for randomly generated obstacle fields of 3, 6, and 9 obstacles.**

***Index Terms*—Quadrotor, Potential Fields, iLQR, Differential Flatness**

I. Introduction

Force potentials have long been employed for navigational path-planning and robot manipulation [2]. In this problem, potential functions are used to generate forces on a vessel or manipulator, pushing the system *downhill* towards a goal, which is the global minimum of the potential function [1]. The main drawback of this method is the possible presence of local minima in a potential field with obstacles. Recently, Kumar et. al. (2022) have demonstrated an algorithm to generate a Rimon-Koditschek (RK) artificial potential to navigate in the presence of ellipsoidal obstacles.

In this paper, we take a slightly different approach and consider the use of velocity potential flow fields for path-planning, which is analogous to the theory of inviscid flow in fluid dynamics. In order to avoid local minima in a potential field made of obstacles, or local velocity stagnation points, we employ a novel and creative technique of modeling obstacle fields with a combination of *elementary flows*. Elementary flows, such as sources, sinks, and vortices, satisfy the Laplace equation for inviscid, incompressible, and irrotational flow. Since the Laplace equation is a linear differential equation, the superposition of these elementary solutions is also a solution to Laplace's equation.

Trajectory optimization methods, such as iLQR, can be sensitive to initial guesses. In order to address this sensitivty, we use velocity potential functions to create a streamline trajectory solution that avoids procedurally-generated random obstacles. Paired with a differentially flat system, the control law required to achieve this streamline can be calculated and can be used as an initial guess to a trajectory optimization algorithm. In this paper, we compare the performance of iLQR with a *warm start* streamline trajectory to the performance of iLQR with a *cold start* initial trajectory.

II. Potential Flow Theory

To compute the streamline used for the warm start, we establish a velocity potential function based on 2-D Cartesian spacial coordinates, $\phi(x, y)$. Any irrotational flow in two dimensions can be completely described by $\phi(x, y)$, where

$$u \equiv \frac{\partial \phi}{dx}, \quad v \equiv \frac{\partial \phi}{dy} \tag{1}$$

and $\mathbf{u}(x, y) = [u, v]^T$ is the velocity field [4].

In terms of $\phi(x, y)$, the condition for incompressibile flow is a form of the *Laplace equation*:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0. \tag{2}$$

In Section IV-B, we describe how uniform flows, source and sink flows, and vortex flows can be used to create a streamline trajectory through an obstacle field. The velocity potential for a uniform flow described by

$$\phi_\infty = V_\infty x \tag{3}$$

where V_∞ represents the freestream velocity in the x direction. Obstacle and goal locations are modeled as sources and sinks, for which the velocity potential is be defined as

$$\phi_s = \frac{\Lambda}{2\pi} \ln r \tag{4}$$

where $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ is the radial distance from the center of the source/sink and Λ is its *strength*. For a repulsive source, $\Lambda > 0$, and for an attractive sink, $\Lambda < 0$. A vortex flow is described by the potential

$$\phi_v = -\frac{\Gamma}{2\pi}\theta \tag{5}$$

where Γ is the strength of the vortex and θ is given by

$$\theta = \arctan\left(\frac{y - y_c}{x - x_c}\right). \tag{6}$$

These elementary flows can then be summed together and differentiated, as shown in Eq. (1), to determine the total velocity flow field, $\mathbf{u}(x, y) = [u, v]^T$ induced by the elementary flows. A streamline ψ represents a particle's trajectory in this steady-state flow, and is defined to be perpendicular to ϕ at every point in the flow field. It can be shown that the slope of a streamline is

$$\left(\frac{dy}{dx}\right)_{\psi=\text{const}} = \frac{v}{u}. \tag{7}$$

Therefore a fluid particle trajectory, or 2D-quadrotor trajectory, can be determined by integrating Eq. (7) from an initial position (x_0, y_0).

III. Two-dimensional Quadrotor

The orientation and state of a quadrotor constrained to the x, y plane can be packaged as

$$q = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}, \quad z = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \tag{8}$$

where x is the quadrotor's East-West position, y is the North-South position, and θ is its orientation angle with respect to the positive x-axis. The non-linear dynamics of the 2D quadrotor, guided by input thrusts of u_1 and u_2 at the left and right rotors, respectively, can be described by the following equation of motion,

$$\ddot{q}(q, u) = \begin{bmatrix} -\frac{\sin\theta}{m}(u_1 + u_2) \\ -g + \frac{\cos\theta}{m}(u_1 + u_2) \\ \frac{a}{I}(u_1 - u_2). \end{bmatrix} \tag{9}$$

The quadrotor's mass is $m = 1$kg, the distance between the rotors is $a = 0.25$m, and the moment of inertia about the center of mass is $I = 0.0625\text{kgm}^2$.

IV. Generating a Warm Start Solution with Potential Flows

A. Modeling the Obstacle Field

In order to create a streamline solution which directs a quadrotor from an initial position $\mathbf{x_0}$ to a final position $\mathbf{x_{goal}}$ and avoids all obstacles, we start by modeling $\mathbf{x_{goal}}$ as a sink fundamental flow to pull streamlines towards the goal. A uniform flow is then added to direct the velocity field from $\mathbf{x_0}$ towards the sink at $\mathbf{x_{goal}}$ and to incentivize moving away from the initial position and towards $\mathbf{x_{goal}}$.

The obstacle field consists of non-overlapping circular obstacles with randomized center locations and radii. The obstacles are generated within set bounds between $\mathbf{x_0}$ and $\mathbf{x_{goal}}$ to create an obstacles field while ensuring no obstacle intersects $\mathbf{x_0}$ or $\mathbf{x_{goal}}$. In addition, the obstacles are generated to maintain a minimum distance between other obstacles in the field.

At each obstacle center, there is a source flow which creates a radial repulsive force to push streamlines away from the obstacle and a vortex flow which creates a circumferential force to curve the streamline about the center of the obstacle. The strengths of the potential flows are determined through an iterative algorithm that first checks for collisions between the streamline and obstacles. Then, the flow strengths are increased for the colliding obstacles until there are no more collisions and the streamline terminates at $\mathbf{x_{goal}}$.

B. Calculating a Streamline Solution

Once the obstacle field is created and a discretized velocity field is calculated, a fourth order Runge Kutta method is used to integrate Eq. (7) from $\mathbf{x_0}$ to $\mathbf{x_{goal}}$. This results in a *streamline trajectory* for quadrotor positions (x, y) within the obstacle field. We then use differential flatness to determine the remaining variables required to define the full state and inputs for the streamline trajectory.

Using cubic splines to define the streamline solution (x, y), the orientation angle θ of the quadrotor along its trajectory can be defined as

$$\theta = -\arctan\frac{\ddot{x}}{\ddot{y} + g}. \tag{10}$$

Furthermore, the input thrust of the left and right motors, u_1 and u_2 respectively, can be defined as

$$u_1 = \frac{1}{2}\left(-\frac{\ddot{x}m}{\sin\theta} + \frac{\ddot{\theta}I}{a}\right) \tag{11}$$

$$u_2 = -\frac{\ddot{x}m}{\sin\theta} - u_1. \tag{12}$$

V. Trajectory Optimization with iLQR

A. Problem Formulation

The trajectory generated by iLQR solves the optimization problem

$$\min_{\mathbf{z},\mathbf{u}} \; l_f(z_N) + \sum_{k=0}^{N-1} \left[l(z_k, u_k) + B(z_k) \right] \tag{13}$$
$$\text{s.t. } z_{k+1} = f(z_k, u_k)$$

where $l_f(z_N)$ is the final cost, $B(z_k)$ is the cost-barrier function due to the obstacle field (ref. Sec. V-B), and $l(z_k, u_k)$ is the LQR cost

$$l(z_k, u_k) = z_k^T Q z_k + u_k^T R u_k. \tag{14}$$

The dynamics constraint $z_{k+1} = f(z_k, u_k)$ is satisfied by the forward pass of the iLQR algorithm.

B. Cost-Barrier Functions

We model *cost-barrier functions* at each obstacle as

$$B(z) = \frac{\alpha}{r^8} - r^4 \exp\left(-br^2\right) \tag{15}$$

where z is the quadrator state, r is the radial distance from the center of an obstacle at (x_c, y_c), and $b = 100$ is a suitable number to achieve a sufficient barrier steepness. The coefficient α depends on the radius of the obstacle r_{obst}, as well as the desired cost, c, on the obstacle's boundary:

$$\alpha = r_{obst}^8 \left(c + r_{obst}^4 \exp\left(-br_{obst}^2\right) \right). \tag{16}$$

Based on visualizations of various iLQR trajectories, we determined that a value of $c = 50$ is sufficient to provide a slight region of safety around each obstacle, and hence $c = 50$ is the constant value that we use for the results presented in this paper.

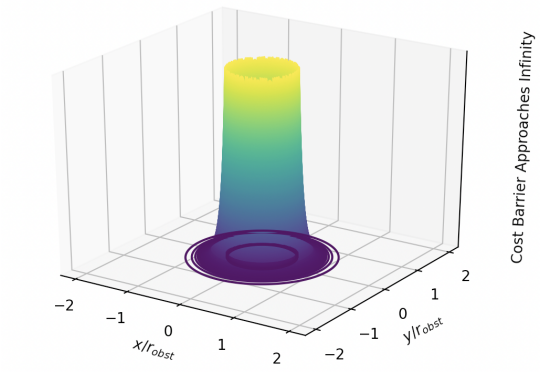

Fig. 1. Cost-Barrier function surrounding an obstacle of radius r_{obst} and center at $x = 0$, $y = 0$.

VI. iLQR Performance Study

A. Study Design

The object of the study is to compare the performance of warm start iLQR to cold start iLQR with regards to convergence rate, convergence time, and final trajectory cost for various obstacle field densities. The warm start uses potential flow path planning to generate an initial guess for iLQR. The cold start uses an initial guess where the trajectory is a linear interpolation between $\mathbf{x_0}$ and $\mathbf{x_{goal}}$ and the control input guess is $u_1 = u_2 = mg/2$ (thrust required to hover in place).

To test the performance, we randomly generate obstacles fields with different densities. $\mathbf{x_0}$ is located at (2,2) and $\mathbf{x_{goal}}$ is located at (28,28). The obstacle field is constrained to a box with corners (5.5,5.5) and (24.5,24.5) and each obstacle has radii between 0.5 and 3 meters. For each obstacle field, warm start and cold start iLQR is applied to the obstacle field and a trajectory solution is attemped. Obstacle fields with 3, 6, and 9 obstacles are tested with 20 trials each for a total of 60 trials.

B. Cold-start Results

In the absence of an obstacle-sensitive initial guess provided by the warm start algorithm, the cold start initial guess is a simple straight line. As can be seen in Figure 2, this may result in an initial trajectory that intersects obstacles between the start position and goal position. Poor initial guesses such as this often increase the runtime and increase the length of the final iLQR trajectory. Poor initial guesses can even result in failure to converge, which is reflected in the disparity in success rate between the two methods (Table I).

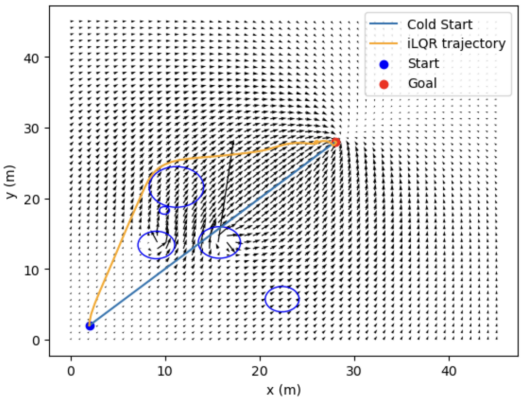

Fig. 2. Quadrotor trajectories using the *cold start* scenario as an initial guess for iLQR.

C. Warm start Results

As expected, the warm start initial guess tends to be close to the final iLQR trajectory. For the same obstacle field with the same cost definitions, iLQR is able to find a trajectory that is more direct and slightly less expensive overall, which can be seen by comparing Fig. 2 and Fig. 3.

D. Discussion

For every number of obstacles, cold start has a lower success rate than warm start. Additionally, as the number of obstacles increases, the success rate of cold start iLQR decrease from 70% to 15%. As the obstacle field becomes more dense, cold start iLQR struggles to converge to a solution. In contrast, the warm start method sees a consistent 85% to 95% success rate that is independent of obstacle field density.

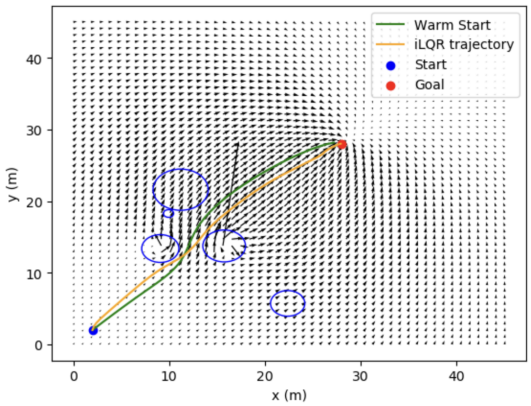

Fig. 3. Quadrotor trajectories using the *warm start* streamline solution as an initial guess for iLQR.

TABLE I
RESULTS OF ILQR WITH WARM AND COLD START INITIAL GUESSES.

	# Obstacles	Avg Success Rate	Avg Cost	Avg Time (sec)
Cold	3	70%	18187	376.5
Warm	3	90%	18204	250.9
Cold	6	35%	27583	586.7
Warm	6	85%	24362	440.9
Cold	9	15%	22689	972.2
Warm	9	95%	24837	462.2

An unexpected result is that the most dense field with 9 obstacles has the highest success rate with the warm start. This may be due to the fact that for dense obstacle fields, the warm start iLQR method chooses a safer but more costly strategy where it avoids the obstacles field altogether, as shown in Fig. 4 and Fig. 5. For less dense obstacle fields, the warm start attempts to navigate through the obstacle field which is riskier and may explain the slightly lower success rate.

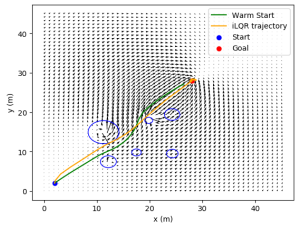

Fig. 4. Warm Start for 6 obstacles.

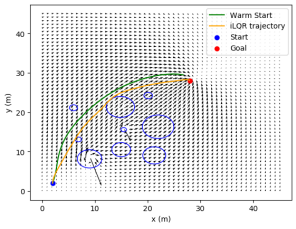

Fig. 5. Warm Start for 9 obstacles.

The average cost of the warm start solution is higher for trials with 3 and 9 obstacles but lower for the 6 obstacle trials. The higher cost of some warm start solution could be due to the initial guess pulling iLQR toward a feasible yet sub-optimal solution.

For all obstacle field densities, warm start converged to a solution faster than cold start. The initial guess to warm start is already feasible and iLQR modifies the initial trajectory to a more optimal solution. This is much faster than cold start iLQR where the initial guess is further away from the final solution and more iterations are required. For both the cold start and warm start case, the time to convergence increases with obstacle field density. However, warm start always converges faster and the gap in run-time between cold start and warm start increases with obstacle density.

VII. CONCLUSION

The results of Table I highlight the promise of using warm start iLQR over cold start iLQR. However, our work has not explicitly addressed the potential of local minima within obstacle-laden potential fields, which is one of the main problems of using potential fields for path-planning [2]. Using the fluid flow analogy rather than the gravitational analogy, these local minima are equivalent to velocity *stagnation points*. Based on visual inspection of the velocity fields and based on all of the simulations that we have run, there do not seem to be any local stagnation points in our obstacle fields. We believe that the presence of point vortices within the obstacles enhances circulation and possibly eliminates the possibility of stagnation points. Future work should include determining whether or not there can be stagnation in points in the velocity fields. Moreover, the theory of potential fluid flow is strictly two-dimensional, so it would be important to explore how this algorithm can be extended to three dimensional path-planning.

SOFTWARE

Our GitHub repository can be found at this link: https://github.com/nruhl25/potential-flow-iLQR

REFERENCES

[1] Pedersen, Morten D., and Thor I. Fossen. "Marine vessel path planning and guidance using potential flow." IFAC Proceedings Volumes 45.27 (2012): 188-193.

[2] Feder, Hans Jacob S., and J-JE Slotine. "Real-time path planning using harmonic potentials in dynamic environments." Proceedings of International Conference on Robotics and Automation. Vol. 1. IEEE. (1997).

[3] Kumar, Harshat, Santiago Paternain, and Alejandro Ribeiro. "Navigation of a quadratic potential with ellipsoidal obstacles." Automatica 146 (2022): 110643.

[4] Anderson, John D. "Fundamentals of Aerodynamics." McGraw-Hill Series in Aeronautical and Aerospace Engineering. (2003).